National Aeronautics and
Space Administration

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

TIMELINE RESOURCE ANALYSIS PROGRAM (TRAP): USER'S MANUAL AND PROGRAM DOCUMENT

JUNE, 1981

(NASA-CR-161812) TIMELINE RESOURCE ANALYSIS PROGRAM (TRAP): USER'S MANUAL AND PROGRAM DOCUMENT (Computer Sciences Corp.) 158 p
HC A08/MF A01 CSCL 09B

N81-26747

Unclas
G3/61 26742

Prepared for:

NASA/George C. Marshall Space Flight Center
Huntsville Computer Complex

Prepared by:

Computer Sciences Corporation
Engineering Systems Department
Project Development and Systems Support Section

"LEGAL NOTICE"

"This report was prepared by the organization(s) named below as an account sponsored by the Tennessee Valley Authority. Neither TVA, the organization(s) named below, nor any person acting on their behalf: (a) makes any warranty express or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or (b) assumes any liabilities with respect to the use of, method, or process disclosed in this report."

(Organization(s) that prepared this report:)

COMPUTER SCIENCES CORPORATION

TIMELINE RESOURCE ANALYSIS PROGRAM
(TRAP)
USER'S MANUAL AND PROGRAM DOCUMENT

JUNE, 1981

Prepared by: Jacqueline D Sessler
Member of Technical Staff B

Reviewed by: David R. Johnson, Manager
Project Development and Systems Support Section

6/4/81, Manager
Engineering Systems Department

ABSTRACT

The Timeline Resource Analysis Program (TRAP) was developed for scheduling and timelining problems. Given an activity network, TRAP generates timeline plots, resource histograms and tabular summaries of the network, schedules, and resource levels. It is written in ANSI FORTRAN for the Honeywell SIGMA V computer and operates in the interactive mode using the TEKTRONIX 4014-1 graphics terminal.

The input network file may be a standard SIGMA V file or one generated using the Interactive Graphics Design System (IGDS). When data is read from a file built on the SIGMA, there are three components of the input: control cards, network description, and resource description. When data is being read from a file built by IGDS, there are five components of the input: Project Definition, Resource Definition, Node Definition, Resource Activity Description, and Predecessor, Successor Linkages.

The timeline plots can be displayed in two orderings: according to the sequence in which the tasks were read on input, and a waterfall sequence in which the tasks are ordered by start time. The input order is especially meaningful when the network consists of several interacting subnetworks. The waterfall sequence is helpful in assessing the project status at any point in time.

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 OBJECTIVE

The Timeline Resource Analysis Program (TRAP) takes information about a project which is read from a data file built from either the IGDS or the SIGMA. Each task of the project is read in detail. Each of these tasks is decoded into an internal format, checked for consistency, and then printed out. Activity durations are scaled according to the number of work shifts per day, a scheduling algorithm assigns start, stop, and slack times to each task, and then resource utilization tables are developed from the completed network. Output plots are selected by the user from menus.

The use of TRAP in timeline analysis differs from PERT analysis. PERT uses an activity-on-arrow formulation which is constrained to a single start and a single stop node. Parallel activities are not allowed in PERT and the graphics representation of a PERT network does not display any meaningful idea of the temporal relationships between activities. On the other hand, TRAP uses a precedence or activity-on-node network formulation which permits multiple start and stop nodes and parallel activities. The timeline, or bar chart, generated by TRAP, displays the temporal relationships between tasks, although it only suggests the precedence relationships.

The capability of TRAP to read a data file built from IGDS was developed for an eventual interface with the system.

The TRAP program was developed for the Computer Services Organization (AH33) in support of the Coal Gasification Task Team (PF15) under Contract NAS8-31640.

1.2 MSFC FORM 3559

See the following page.

USER RESOURCES PROJECTION PLAN
HUNTSVILLE COMPUTER COMPLEX

1. NAME OF REQUESTER (24 characters):		4. NAME OF PROJECT (30 characters):
Tom French		TVA Coal Gasification
2. TELEPHONE NUMBER: 453-4024	3. ORGANIZATION (4 characters): PA01	5. PROJECT (UPN) (7 characters): 778-44-29
		6. DATE OF REQUEST: 8/8/80

JOB NUMBER	SERIES CODE	COG. ACT.	PROJECT	SYSTEM
	—	PA —	778 —	R5002 1

7. JOB TITLE (24 characters): TVA Coal Gasification

9. JOB DESCRIPTION (300 characters):
Provide flow sheet graphics; absorption, stripping, and distillation column design; facility and equipment sizing and plant layout; startup procedure timelines; and analyses of: cost and economics, material and energy balance, combustion equilibrium for the TVA Coal Gasification facility.

19 81 FISCAL YEAR RESOURCES PLAN

	ESTIMATED HOURS/SUPS					COST
	1ST. QTR	2ND. QTR	3RD. QTR	4TH. QTR	TOTAL	
LABOR	10. 1926	11. 1927	12. 1926	13. 1927	14. 7706	15. $105,957.50
COMPUTER	16. 10	17. 11	18. 10	19. 11	20. 42	21. $5,880.00

23. COMPUTER IDENTIFICATION: UNIVAC 1100

22. TOTAL COST: $111,837.50

24. FUNDING SOURCE: ☐ R&PM ☐ D&O ☐ R&D ☒ REIMBURSABLE

25. WORK PACKAGE:	26. TASK NUMBER	27. TASK TITLE (24 characters):
34X		

28. NAME OF PROGRAMMER/ANALYST (24 characters)	29. NAME OF MANAGER (24 characters):	30. ORGANIZATION:
Mike Fague	Mike Fague	653
31. NAME OF MONITOR (24 characters):	32. TELEPHONE NUMBER:	33. ORGANIZATION:
R. Martin	3-2294	AH33

34. COMMENTS:
Scientific/Engineering

ORIGINAL PAGE IS OF POOR QUALITY

☐ OVERGUIDELINES

36. SIGNATURE OF AUTHORIZED REQUESTER: Rice H. Toneel

37. DATE RECEIVED:

Form 3550 (Rev. July 1978)

BEFORE COMPLETING THIS FORM, CONTACT ADP [illegible]

1.3 BACKGROUND INFORMATION

There exists the Activity and Resource Scheduling Program, or SKILL, which was originally developed to perform time-in-motion analysis and activity timelining for mission operations analysis. The SKILL program was developed and resides on the UNIVAC 1100 series.

After a study of the capabilities available with SKILL it was decided that if it were put on the Honeywell SIGMA V computer and some modifications and enhancements were made to it, it would support Coal Gasification requirements. This modified program is the Timeline Resource Analysis Program.

2.0 PROBLEM TASK DESCRIPTION

The data file which TRAP has the capability of reading will originate from a network drawn by a user in the form of nodes and arrows. The data file built on the SIGMA is built in the text editor. Once a data file is read, TRAP has the responsibility of putting the information into the form of tables and bar plots for the purpose of scheduling.

TRAP has the capability to output six different tables and three different plots. There are two Echo Reports used to check accuracy of the input. The Predecessor-Successor Table shows the flow of the network as it is input, whereas the Task Scheduling Table gives the tasks in order of time which assists the user in scheduling on time and in order of importance. All of the tasks in these tables are shown by their code except for the Full Title Table which shows them in waterfall order by title rather than code. The last of the tables available is the Resource Histogram Table which shows the amount of a resource needed at any one time.

The three plots available for output are two bar charts for task flow in order of input or by start time and a resource histogram plot. The plots make it easier to see the full flow of the network. The picture representation is also helpful in a presentation for an overall view.

3.0 METHOD OF SOLUTION

To build a data file on the SIGMA V Computer, the user must input the data through the text editor in the proper format. When a file is built on IGDS, it will be using a PDP Computer. When using the interface with IGDS, the user inputs the nodes and links graphically. As a node is input graphically, a menu of sorts will be printed to the screen for the user to input particular characteristics about that node. The linkages for each node will be input graphically after all of the nodes and resources have been defined. After the information has been input to the screen on the IGDS side, a program will reformat the data and build a sequential data file. The data will then be transferred to a file in the Honeywell SIGMA V Computer.

The data is then in a format that the user on the SIGMA can use. TRAP uses a menu where the user determines the flow of the program. The user must first define and load the network source file. The network must then be processed before the user can continue. At this point, the automatic hardcopy, the segmentation of plots and tables, or which plot or table required is chosen.

Figure 3-2 shows the flow of data when interfaced with IGDS. When data files are built on the SIGMA the flow is only within its own data file and output, though it is not the same data file built from IGDS.

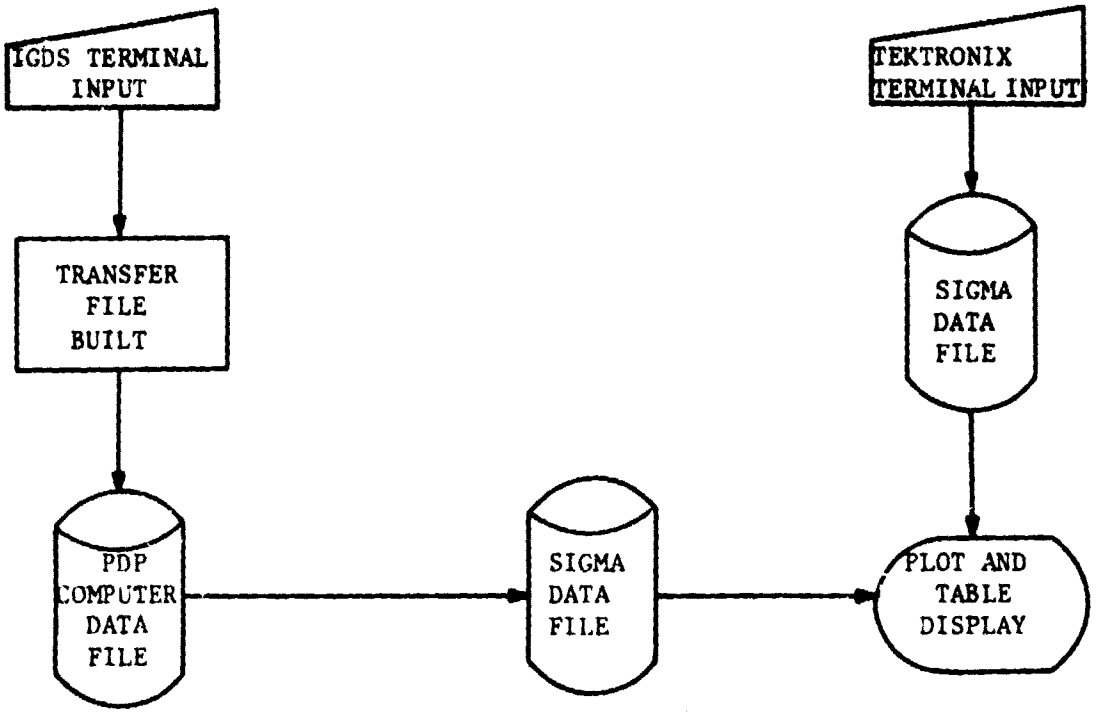


FIGURE 3-1

FLOW OF TRAP DATA

Network processing is the first of the major capabilities of the program. The data read from the data file includes a list of the tasks' predecessors, if any, the nominal duration of the task, the number of work shifts per day, a list of the resource seize/release actions and certain flags. Once the entire network file is read, the program builds a list of successors for each task. The complete predecessor-successor table allows the program to easily scan the network in either direction. A listing of the complete predecessor-successor table is available on output.

Before the scheduling sequence can be determined, the activity durations must be assigned. The duration is then scaled according to the work schedule. TRAP will scale the activity duration as the task is declared to have one, two, or three shifts per day. The shift scale factor is the ratio of hours per week (168) to the actual number of work hours (40, 80, or 168) which gives factors of 4.2, 2.1, or 1.0 for the stated work weeks.

The next step is to develop a topological ordering of the tasks from the predecessor-successor relationships. Basically, the ordering algorithm starts with the first task it finds which has no predecessors, then works forward through the network following the chain of successors. When it has worked all of the way through, it reverses and follows the predecessor chains looking for branches which were not previously traversed. In this way it scans back and forth until all of the tasks have been traversed.

After the topological sort, start and stop times are assigned to each task. The first task with no predecessors is arbitrarily assigned a start time of zero. All of the subsequent tasks which were traversed on a forward pass are assigned start times equal to the stop time of its latest scheduled predecessor. Tasks traversed on a backward pass are assigned start times equal to the start time of its earliest scheduled successor minus the duration of the task being scheduled.

To determine the absolute slack time associated with a task, the entire scheduling process is repeated beginning with a backward pass. The absolute slack time for any task is then the magnitude of the difference between the start times assigned to the task by each schedule. Tasks on a critical path will have the same start time regardless of the scheduling sequence; hence, their slack time will be zero.

After all times are assigned, TRAP will subtract the most negative start time from all of the start and stop times.

In this way, the earliest task is scheduled at time zero. If, on the other hand, it is desired to display task time relative to a given task, the task may be flagged on input, causing its start time to be subtracted from all of the start and stop times, giving it a start time of zero. Another available flag allows a value to be taken as the absolute start time for the network and all other times are biased to it.

As one final scheduling option, tasks may be flagged on input to signify that if the task is not on a critical path, it should be scheduled either as early or as late as possible within its slack time.

The Resource data read into TRAP includes the resource seize/release actions for each task mentioned earlier as well as data read from the resource block. The resource data includes a sum check flag, a value type flag, a function index, and an initial value for each resource. The seize/release action includes a code to determine if a resource is being seized and/or released and the quantity.

TRAP builds a table for the resources. At the time that the seize/release actions are being read for the table, it checks the value type flag to determine if the amount is an integer or a floating point and it also checks for a function index for a special equation to determine the amount seized/released. After all of the resources are seized and released, the sum check flag is checked to see if it is desireable to make sure that the resource is at zero. The table is a single dimension array which has the maximum level, number of levels, and level and time when there is a change in the level for each resource. From this table, the resource histogram tables and plots are output. At the time of output for the plots, the initial value is checked and a line is drawn across the plot at that level so that it can be seen if the amount of resources needed or used is correct.

4.0 PROGRAM DESCRIPTION

4.1 OPERATING ENVIRONMENT

4.1.1 Hardware

- Computer - Honeywell SIGMA V
- Core Requirement - 61.2K words
- Magnetic Tapes - not required
- Card Punch - not required
- Plotter - not required
- Drum/Disk - 50 Granules
- Other - not required

4.1.2 Software

- Operating System - Honeywell SIGMA V
- Control Program - Five (CP-V)
- Programming Language - ANS FORTRAN
- Type of Run - Demand
- Library Subroutines - LXLIB:ANS.NUGRAF

4.2 PROGRAM SPECIFICATIONS

TRAP was developed on the SIGMA V for the purpose of eventually interfacing with IGDS and to increase the number of tasks per network. It has been developed to be able to read a data file built from IGDS and output specified scheduling plots and tables. TRAP is now able to handle up to 300 events as well as segment a large network in order to analyze a portion of it. It also now has an option to be able to schedule an event early or late. The information on resources has been expanded in order not to be restricted to too general a type of information.

4.3 SUBROUTINES

MAIN Routine - This routine has the main menu which allows the user to determine the program flow.

ASK - This routine asks for the title and units for the network.

AUTOC - This routine sets automatic hardcopy ON or OFF.

BAR - This routine sets up loops for drawing bars and triangles and determines if a page is filled.
Calling Sequence - MARK, EARLY, LATE
MARK - used to mark if a task has been scheduled or not during the scheduling process.
EARLY - flag value for tasks scheduled early.
LATE - flag value for tasks scheduled late.

CKHIST - This routine determines which parts of the Histogram plot belong in the segment.
Calling Sequence - TIMES, TIMES2, ISET, NLVL, J
TIMES - the time of a resource being checked and compared to see if it's within the segmentation boundaries.
TIMES2 - the next change in a resource level to be checked to determine if it's within the segmentation boundaries.
ISET - set to 1 or 2 depending on if this task is within segmentation boundaries or not.
NLVL - the number of different levels during the use of a resource.
J - which level is being tested.

CKHTAB - This routine checks the times on the histograms to determine if they are within the segment boundaries for the histogram table.
Calling Sequence - HTABLE, IS, IE, WORDRF, WORDRL
HTABLE - array of times and levels of a resource.
IS - subscript for HTABLE; the first time in a line of print.
IE - subscript for HTABLE; the last time in a line of print.
WORDRF - the earliest time of the segmentation.
WORDRL - the latest time of the segmentation.

CKSTMP - This routine sets start and end times for tasks within segment boundaries for waterfall plots.
Calling Sequence - WORDRL, WORDRF, ISKP
WORDRL - same as above
WORDRF - same as above
ISKP - flag set if a task is to be skipped for output because it's outside the segment boundaries.

CKSTTM - This routine checks the time to determine if the task is completely out of the boundaries of the segment for waterfall type tables.
Calling Sequence - STIME, IQT, TSTOP, WORDRF, WORDRL
STIME - start time of task.
IQT - flag set if the task is outside segment boundaries.
TSTOP - stop time of task
WORDRF - same as above
WORDRL - same as above

CKTSKN - This routine determines if the task is within the segment boundaries for the timeline type tables.
Calling Sequence - TASK, IQT, IQUTS, IORDRF, IORDRL
TASK - the task being checked.
IOT - flag set if task is not within segment boundaries.
IQUTS - flag set depending on if the task is outside or inside the boundaries.
IORDRF - the first task in input order to be inside the segmented boundaries.
IORDRL - the last task in input order to be inside segmented boundaries.

CLEAR - This routine clears the screen and resets the character size.
Calling Sequence - ISIZE
ISIZE - integer to determine character size

COPY - This subroutine copies the entries from one doubly linked list to another.
Calling Sequence - LIST1, LIST2
LIST1 - the array containing the links for the chain to be copied.
LIST2 - array containing the links for the chain to which the first chain is to be copied.

CORNER - This routine generates a list of all of the start and stop times.

DEL - This routine will delete a specified node from a doubly linked list.
Calling Sequence - NODE, LIST
NODE - pointer to entry to be deleted
LIST - array from which entry is to be deleted

ECHO - This routine prints an echo report of the input data.
Calling sequence - MM, IPCESS
MM - integer to flag which report within the routine is desired
IPCESS - an integer to flag if the network has been processed yet.

EDGSET - This routine builds the predecessor list by taking the information from the edge definition block. This block defines the tasks links and then defines its predecessor and successor.
Calling Sequence - EDGPR, EDGSU
EDGPR - This is the predecessor of the link defined.
EDGSU - This is the successor of the link defined.

EDIT - This routine was added for the future which allows the user to edit a data file that was built on the SIGMA without returning to the text editor.

FRAME - This routine draws the frame, tic marks and labels for the bar plots.

HIST - This routine builds the histogram table for resources.

HISTO - This routine sets up arrays for the histogram plots to be output.
Calling Sequence - MM, IPCESS
MM - same as above
IPCESS - same as above

HPLOT - This routine draws and labels histogram plots.
Calling Sequence - TIMES, NLVL, INMEN, INDMEN, IPASSZ, ICOUNT, IIUNIT, ISEG, WORDRF, VINIT
TIMES - array with all times and levels of a resource
NLVL - same as above
INMEN - integer value of maximum level of a resource
INDMEN - integer value of mid-point of level of a resource
IPASSZ - counter to keep track of the number of resource plots on a page
ICOUNT - total number of resource plots to be output
IIUNIT - time units of network in hollerith
ISEG - flag set if segmenting network
WORDRF - same as above
VINIT - initial value of a resource

INERR - This routine checks for and prints errors from input.

INIT - This routine initializes the terminal routines.

INPUT - This routine reads data from data files built on the SIGMA.

INPUT2 - This routine reads the data from a file built on the IGDS.

INSERT - This routine puts a task into an array to be used as a working chain for putting tasks in the correct order.

Calling Sequence - NODE, LKSUCR, LIST
NODE - task that is to be inserted onto array
LKSUCR - point in array that task will be inserted
LIST - array being built

NTWKPS - This routine is the driver for the forward and backward pass through the network to reorder tasks in the waterfall order.

Calling Sequence - IPASS
IPASS - integer value to flag pass as early or late

PLOTBR - This routine determines where and writes task title, and plots bars.

Calling Sequence - MARK, EARLY, LATE
MARK - same as above
EARLY - same as above
LATE - same as above

PRNTS - This routine prints out all of the tables except echo and full title tables.

Calling Sequence - MM, IPCESS
MM - same as above
IPCESS - same as above

PROCESS - This routine in the main driver for getting the network processed for output.

Calling Sequence - IPUT
IPUT - an integer to flag if the network has been read

PRTNET - This routine prints the network task description (titles) in waterfall order with each task followed by the descriptions (titles) of its predecessors and successors

RESRCE - This subroutine performs error checking on the resource seize/release table and counts the number of resources.
Calling Sequence - PRNTNW
PRNTNW - integer flag set if error messages are to be printed to the screen.

SCHED - This routine gives a start and stop time to each task going forward and backward through the network.
Calling Sequence - IPASS
IPASS - same as above

SCLY - This routine determines the maximum scale of Y-axis on resource graphs.
Calling Sequence - INMEN
INMEN - same as above

SEGMNT - This routine sets the segment boundaries if segmentation of a plot is desired.

SEQNCE - This routine completes the precedence/successor sequences.

SETUP - This routine does the setup for plotting a timeline graph.
Calling Sequence - IK
IK - the number of the task that is being set up.

SORT - This routine sets start times in correct order for waterwall order to be inserted into list.
Calling Sequence - NODE, IPTR, VALUE, TIE, LIST, LHEAD
NODE - the number of the node (task) that is being inserted
IPTR - the point in the array that the start time will be inserted
VALUE - start time of node to be inserted
TIE - start time of node to be compared with
LIST - array being built
LHEAD - index of the stored corner chain header record

TIMLIN - This routine computes time scale for input and waterfall order plots.
Calling Sequence - MM, IPCESS
MM - integer to flag which plot within the routine is desired
IPCESS - same as above

TRNGLE - This routine determines where and writes task title and plots triangles.

WAIT - This routine stops execution after a page of output until the user decides to clear the page. An option may also be chosen prior to output which will cause this routine to automatically get a hardcopy before the page is cleared.

Calling Sequence - I, ISIZE, ICPY

I - is set to 0 for counting of lines on the page following the one being cleared.

ISIZE - code for size of the print

ICPY - flag set if tables and plots are to be automatically copied

WORK - This routine processes a pass forward and backward working its way through the network until all of the tasks are marked as being scheduled.

Calling Sequence - IMARK, NLINK, LIST, MLINK

IMARK - integer array used to mark tasks during scheduling process.

NLINK - number of predecessors or successors (depending on if it's a backward or forward pass successively)

LIST - the array of predecessors or successors (depending on if it's a backward or forward pass successively)

MLINK - array of number of predecessors or successors (depending on if its a forward or backward pass, successively)

IHALFL - This function takes a word and shifts it right one half word, so the left half can be read.

Calling Sequence - IARG

IARG - word to be shifted.

IHALFR - This function takes a word and shifts it left half a word and then back right half a word, so the right half can be read.

Calling Sequence - IARG

IARG - same as above

ISTRL - This function takes two words and shifts both to the left a half word, shift the second back to the right and adds the two together. This puts the right half word of the first word into the left half of the second.

Calling Sequence - IFM, ITO

IFM - word to be shifted and added to left half of another word

ITO - word to be shifted twice so another word can be stored in its left half.

ISTRR - This function takes the right half of one word and the left of another and puts them in one word.

Calling Sequence - IFM, ITO

IFM - word that right half is stored

ITO - word that left half is stored

RSCFCN - This function calculates special seize/release quantities that the user may choose.

Calling Sequence - ITSK, IFCN, IRSC, TCOR

ITSK - task seize/release is being calculated on

IFCN - integer to specify which function to use for the calculation

IRSC - the number of the resource for the task

TCOR - start time of task comparing

5.0 OPERATING INSTRUCTIONS

5.1 DECK SETUP

Before TRAP can be executed, the user must know the name of the file where the data is that is to be used. The user must also know the title for the run, and the units for the run.

For access to the SIGMA V Computer, the user must do the following.

> Dial a number assigned by the SIGMA operators. When a high pitched tone is heard, the white button is to be pulled up. It should respond with the date and other information and then respond with:
>
> LOGON PLEASE
>
> At this point the user inputs the account number assigned to him. Some additional information will be displayed to show that the user is now on.
>
> The user whould then hit the escape key and then the E. This is done to suppress duplicate printing of characters.

To set data up to be read input:

```
PLATEN 140
SET F:1 UC
SET F:5 DP/DATF
```

where DATF is the data file name being used.

To start execution input:

START EXECUTE

5.2 INPUT

5.2.1 Cards

The following are card images that are input through the editor for data files built on the SIGMA.

Card Type 1 - Table Type

FIELD	COLUMNS	DEFINITION
1	1-2	Defines if at the beginning of a table (TA), at the end of the data (EO), or a comment card (C).
2	7	When at the beginning of a table, determines table type: F - Task Description Table R - Resource Description Table

Card Type 2 - used for special start time

FIELD	COLUMNS	DEFINITION
1	30	Special start time flag: X
2	33-37	Start time (Real # decoded to F5.0), optional

Card Type 3 - Task Description

FIELD	COLUMNS	DEFINITION
1	2-5	Task code - internal identification of task (character)
2	6-29	Task description - title for identification on plots and tables
3	30	Critical task flag Z: time will be at Ø when this task starts, optional.
4	31	Determines if task will be forced scheduled early or late. L - if not late, forced late BLANK - if not early, forced early Optional

FIELD	COLUMNS	DEFINITION
5	32	Shift option determines the number of shifts per day. Default is 3.
6	33-37	Duration of task. Real number decoded to F5.1.
7	38-53	Up to four predecessor tasks identified by their task codes.
8	54-77	Up to three seize/release activities within each activity there are three fields:
8A	54,62,70	Seize/release code: + - Seize - - Release * or blank - seize and release
8B	55-59, 63-67, 71-75	Number of units seized or released (integer)
8C	60-61, 68-69, 76-77	Resource title code to be described in resource table (character)

Card Type 3A - First Task Description continuation

FIELD	COLUMNS	DEFINITION
1	1	Continuation number: 1
2	38-53	Predecessor tasks 5-8 identified by task codes
3	54-77	Seize/release activities 3-6. For breakdown of each activity, see type 3, Fields 8A-8C.

Card Type 3B - Task Description Continuation Card

FIELD	COLUMNS	DEFINITION
1	1	Continuation number: 2
2	38-53	Predecessor tasks 9-12 identified by task codes
3	54-77	Seize/release activities 7-9. For breakdown of each activity, see type 3, fields 8A-8C.

Card Type 3C - Task Description Continuation Card

FIELD	COLUMNS	DEFINITION
1	1	Continuation number: 3
2	54-77	Seize/release activities 10-12. For breakdown of each activity see type 3, fields 8A-8C.

Card Type 4 - Resource Description

FIELD	COLUMNS	DEFINITION
1	2-3	Resource title code as used on task description card for internal identification
2	5-28	Resource description to identify on plots and tables
3	29	Sum check flag to determine if resource seize/releases should be checked to see if it ends at the level of zero. Integer variable of 1 to check and defaults to 0 for not to check.
4	30	Value type flag to determine if the resource will be in floating point, integer of 1 or an integer which is the default of 0.
5	31	Function index: An integer value which defaults to zero or is an index to determine a special equation to determine resource utilization
6	32-36	Initial value of the resource
7	37	Constrained resource flag. Integer value set to 1 if resource is conconstrained and defaults to 0.
8	38-42	Constraint level of the resource.

5.2.2 Drum/Disk

Data files built from IGDS will be kept and read from a disk. The current format specifications at the time of documentation are as follows:

RECORD			FORMAT
1	LENGTH OF PROJECT DEFINITION BLOCK (PDB)		
2	LENGTH OF RESOURCE DEFINITION BLOCK (RDB)		
3	LENGTH OF STATIC PART OF NODE DEFN BLOCK (NDB)		5I6
4	LENGTH OF RESOURCE ACTIVITY DESCRIPTOR (RAD)		
5	LENGTH OF EDGE DEFINITION BLOCK (EDB)		
6	POINTER TO 1ST RECORD WITH RESOURCE DATA	-	I6
7	PROJECT TITLE (BEGIN PDB)	-	9A4
8	NETWORK ID	-	A4
9	NETWORK TITLE	-	9A4
10	NETWORK LEVEL	-	I1
11	START TIME OF UPPER LEVEL NODE	-	F10.1
12	UNITS OF TIME FOR NETWORK	-	2A4
13	NETWORK DURATION	-	F10.0
	POINTER TO 1ST RECORD W/NODE DATA	-	I6
+1	# OF RDB'S TO FOLLOW	-	I6
+2	RESOURCE ID CODE	-	A2
+3	RESOURCE DESCRIPTION	-	6A4
+4	SUM CHECK FLAG	-	I1
+5	VALUE TYPE FLAG	-	I1
+6	CONSTRAINED RESOURCE FLAG	-	I1
+7	CONSTRAINT LEVEL	-	F10.0
+8	INITIAL QUANTITY OF RESOURCE	-	F10.1
+9	FUNCTION INDEX	-	I2
	(NEXT RDB ENTRY)		

	POINTER TO 1ST RECORD W/EDGE DATA	- I6
+1	POINTER TO NEXT NDB (0 THIS IS LAST NDB)	- I6
+2	NODE ID	- A4
+3	NODE DESCRIPTION	- 6A4
+4	SHIFT FACTOR	- I1
+5	DISCRETE FLAGS (SEE * BELOW)	- 4I1
+6	NODE DURATION	- F7.1
+1	# OF RAD'S TO FOLLOW	- I6
+2	RESOURCE ID CODE	- A2
+3	RESOURCE ACTIVITY CODE	- A1
+4	RESOURCE ACTIVITY QUANTITY	- F5.1
	(OTHER RADS FOR THIS NODE)	
	BEGIN NEXT NDB	
	# OF EDB'S TO FOLLOW	- I6
+1	EDGE ID	- A4
+2	PREDECESSOR NODE FOR THIS EDGE	- A4
+3	SUCCESSOR NODE FOR THIS EDGE	- A4
	EDGE ID (NEXT EDB)	

* I1 - SPECIAL START TIME OR CERTAIN TASK @ TIME 0
I1 - TASK FORCED EARLY OR LATE

5.2.3 Other

A master menu will display options for the user to choose. Before anything can be done, number one in the menu must be chosen. This reads the data file into core for the program's execution. The next option that must be chosen is number three. This will process the network by determining its waterfall order. At this point, any of the other options may be chosen.

If option two is chosen here, any errors that may have occurred during one and two will be printed out. This is advantageous in saving time so the user won't be getting tables and plots of no help. Option four is the automatic hardcopy. This saves time, because it allows the user to set the program for a lot of output and leave the area. When this option is chosen, a message comes to the user at the terminal and states if the automatic hardcopy is on or off. It then asks the user to input a 1 if it is to be on or a 2 if it is to be off. It then returns to the main menu. Option five is another special option. It sets a segmentation flag. When this option is chosen, a message informs the user if the flag is on or off. It then asks the user to input a 1 if it is to be on or a 2 if it is to be off. If the flag is off, it returns to the main menu, otherwise, it asks four questions about the segmentation boundaries. The first asks for the first task of the segment for an input-order table and plot with the second question asking the last task of the same. The third question asks for the earliest time of the segment for a waterfall-order table and plot and the last question asking the latest time shown for the same. It then returns to the main menu.

Option six through eighteen are chosen depending on the output needed. A particular table or plot may be chosen, all tables or all plots may be chosen and all tables and all plots may be output.

When option fifteen is chosen, which outputs resource histogram plots only, another menu is displayed on the screen. This menu furnishes three options. All resources may be output, selected resources may be output or no resources may be chosen which returns it back to the main menu. If the option to graph selected resources is chosen, when it is stated on the terminal, an X may be input to list all resources so a choice may be made as to which will be graphed. The program will then ask how many resources are wished to be output. Up to three can be output at once. The program then asks for the indices for the desired resources. At this point, the user should input a single integer number of the resource desired followed by hitting return. As stated earlier, this can be

done three at a time. After this is done, it returns to the menu stated at the beginning of this paragraph.

The last option is zero which will terminate the program.

There is an option to edit the data, but at the time of documentation, this feature is not yet available.

5.3 OUTPUT

TRAP has the capability to output six different tables and three different type plots. The terminal being used will have graphic capabilities to output the plots. The total number of pages output depends on the size of the network and how many tables and plots are required. For each task table, fifty tasks will be output per page, while with the resource table, six different resources can be output per page. For timeline plots there will be forty tasks per page and on the resource histograms, there are three resources per page. (See test case for samples of output).

5.3.1 Printout

The first tables available are the Echo Reports. One for the network, the other for the resources. These echo back what was input by numbering the tasks, printing its label, title, and duration. It also prints the total number of predecessors per task, prints up to the first five by label, and also prints the total number of seize/releases by task and the first five as input in data file. The resource echo report prints all of the resource codes and titles.

The next table available is the Predecessor-Successor Table. It prints out each task label and title and up to the first seven predecessors and successors associated with it.

The next table is the Task Scheduling Table. This prints the task title along with its duration, slack time if it's not on the critical path, start time, and end time. It prints these in the order of time that it is scheduled rather than in the order that it was input.

The fifth table available is the Full Title Table. This table gives the same information that the Predecessor-Successor Table gives except the task title is used in place of the task code for each predecessor and successor, and it is in waterfall rather than input order.

The last tables are the Resource Histogram Tables. There is a table for each resource which gives the time and level each time there is a change in the amount of that resource being used. It also gives the maximum level that the resource has reached, the initial value of the resource, and shows if the constrained resource is on or off.

5.3.2 Plots

The first of the plots output is a waterfall plot. This is a bar chart with each task plotted in order of the time that each begins. The bars are either solid or partially cross-hatched. The solid bar represents a task on the critical path. The cross-hatched section of the other bar represents the duration of the task while the plain section is the slack associated with it. The duration is also given with each bar. A triangle is shown for those tasks that have no duration.

The second plot is also a bar chart with each type bar representing the same as above. This plot is different in that the tasks are plotted in the order that they were input from the data file.

The last plots available are the histogram plots. There is a plot for each resource that shows the level of resource being used during a period of time. There is also a horizontal line through each plot to show the initial value of each resource.

5.4 RESTRICTIONS/LIMITATIONS

TRAP has a few restrictions that the user should be aware of. They include the following:

Maximum number of tasks per network - 300
Maximum number of predecessors or successors - 12
Maximum number of seize/releases per task - 12
Maximum number of resources - 50
Maximum number of characters for title of the network - 36
Maximum number of characters for units - 6

5.5 DIAGNOSTICS

TRAP will automatically print error messages for the user on most errors that could be made on input. It will also distinguish between a fatal error and a warning. A fatal error will cause the execution to terminate while with a warning, execution will continue.

Errors on the tasks at input include the following:

FATAL - more than one critical job
FATAL - more than one start time
FATAL - critical job and start time

These can be corrected by looking in column 30 of the data in the file. There should be only one X or one Z for each network and both cannot be in one network.

FATAL - number of tasks exceeds available storage
FATAL - Task 'XXXX' has more than three continuation cards

To correct the first, some of the tasks need to be taken out. This shows that there are more than 300. For the second, XXXX will be a task code. There can be no more than three continuation cards per task because of the limited amount of storage. The last error will also occur if something other than a 1, 2, or 3 is in column one.

Errors on the resources at the time of input include the following:

WARNING - Resource code 'AB' is undefined.
Referenced by task 'XXXX'.

To correct this, the user should look at a task with code 'XXXX' in columns 60-61, 68-69, or 76-77 in the input data file. The resource code 'AB' will be found but it should also be found in columns 2-3 of the resource table in the data file. If this warning is displayed, execution will continue and the program will create a resource for the code.

FATAL - number of resources exceeds available storage

To correct this, some resources need to be taken out. This shows that there are more than 50.

FATAL - invalid seize/release code 'x' in task 'XXXX'

To correct this, the user should look at task 'XXXX' in columns 54, 62, and 70 in the input data file. The only codes that should be in these columns are +, -, *, and a blank.

WARNING - flow does not provide zero - sum seize
releases for resource 'AB', NET = '123'

This warning is saying that the network has seized (+) and released (-) resource with code 'AB' and at the finish of the network, the resource is not back at zero. Instead, it came to '123'. To get rid of this warning, the amounts seized and released will need to be changed or the sum check flag needs to be changed to 0.

While the program is in the process of putting the network in waterfall order, the following errors may occur:

FATAL - invalid predecessor.
Task 'XXXX', Pred label 'YYYY'

To correct this error, the user should check to make sure that label 'YYYY' is an actual label of a task in the network..

FATAL - Task 'XXXX' has more than the maximum number of successors allowed

The maximum number of successors allowed per task is 12. To correct this, the user will probably have to put in a couple of dummy tasks.

At the time when the program is asking for the number of resource histogram plots the user wishes to output, the next error may occur:

ERROR - '12' is not a valid resource index

This error shows that when the program asked for indicies for the desired resources, the user input a number larger than the number of resources input.

There is one error that may occur on input that will not result in an error message. When the program does not continue as it should when option 1 is chosen from the menu by returning to the main menu after the network title and unit were input, the program has gotten caught in a loop. The user should check the network that was input. When task 'AAAA' has a predecessor 'BBBB' and task 'BBBB' has a predecessor 'AAAA' this will cause a continuous loop. The user will have to force a termination to get control back.

The last set of errors that may occur will happen when choosing certain options of the main menu in the wrong order. The following errors may occur:

DATA HAS NOT BEEN READ; HIT RETURN AND CHOOSE A '1' IN THE MENU.

This shows that the data has not been read in so it can't be processed. Follow the directions given to correct it.

THERE ARE ERRORS IN THE NETWORK; HIT RETURN AND CHOOSE A '6' IN THE MENU.

This shows that a fatal error or warning has occurred. To display the error, the user should follow the directions given.

DATA HAS NOT BEEN PROCESSED: HIT RETURN AND CHOOSE A '3' IN THE MENU.

No tables or plots can be output without the information received through the process. To correct this, follow the directions given.

BEGINNING AND ENDING TASK NUMBERS WERE NOT GIVEN, AND SEGMENT FLAG IS TURNED ON.

This error is given only when an option is chosen for a timeline order table or plot. The user chose segmentation but did not input the task codes to be the segment boundaries. The program will skip the plot or table needing the information.

BEGINNING AND ENDING START TIMES WERE NOT GIVEN AND SEGMENT FLAG IS TURNED ON.

This error is given only when an option is chosen for a waterfall order table or plot. The user chose segmentation but did not input the starting and ending segment boundaries. The program will skip the plot or table needing the information.

5.6 TEST CASES

The following are two test cases. The first is a network that is all on the critical path and it has resources associated with it. The second has no resources, but does have some slack time associated with it. The data files as input through the editor are also included.

```
1 000 TABLE F-
2 000 C
3 000  0000START                          0 0
4 000   191191 AWARD A/E CONTRACT         3 50000
5 000   192192 PLANT CONCEPT ENG PK       2 0 191
6 000   193193 60% PROCESS PLANT PK       6 0 192
7 000   211A211A COMPONENTS LIST           1 10000                    *  8 0LA*  5 0JO
8 000   211B211B SOU-COMP DATA COLLE       1 0213A                    *  8 0LA*  1 0MI
9 000   212A213A PROC CONT-COMP D CL       2 0211B                    *  5 0LA
10 000  212B212B CONTRACT-COMP D COL       6 0212A                    *  3 0LA+ 17 0C8
11 000  213A213A RAM METHODS REVIEW        1 00000                    *  8 0RA*  1 0MI
12 000  213B213B SOU-RAM MODEL             1 0213A                    *  8 0RA
13 000  213C213C PROC CONT-RAM MODEL       2 0213B                    *  5 0RA
14 000  213D213D CONTRACT-RAM MOD DU       4 0213C                    *  1 0RA+ 19 0C8
15 000  214A214A INITIAL RAM ASSESSM       4 0213D 192                *  7 0RA*  3 0MI*  5 0JO
16 000  214B214B 60% DESIGN QUAL ASM       1 S214A 193                *  8 0RA*  2 0MI*  5 0JO
17 000  214C214C 60% DESIGN QUANT AS       4 0214A 193                *  8 0RA*  3 0MI*  5 0JO
18 000  214D214D CONTRACT-RAM MOD AS      11 0213D                    *  1 0RA+ 13 0C8
19 000  215A215A DYNAMICS MODEL DEFI       1 00000                    *  8 0DE*  2 0MI
20 000  215B215B SOU-DYNAMICS MODEL        1 0215A                    *  8 0DE
21 000  215C215C PROC CONT-DYNAM MOD       2 0215B                    *  5 0DE
22 000  215D215D CONT-DYN MODEL DEVE       4 0215C                    *  1 0DE+ 14 0C8
23 000  216A216A INITIAL DYNAMICS AS       4 0215D 192                *  8 0DE*  2 0MI
24 000  216B216B 60% DESIGN ASSESSME       6 0216A 193                *  8 0DE*  3 0MI
25 000  216C216C CONT-DYN MODEL ASSE      14 0215D                    *  1 0DE+  8 0C8
26 000  216D216D DESIGN RECOMMENDATI       2 00183                    * 10 0DE*  5 0MI
27 000  221A221A COMPLETE DESIN TOOL       2 00000                    +  6 0C8*  7 0ST* 10 0GO
27 100 1                                                             *  10 0SU*  3 0KE*  3 0CO
28 000 2                                                             *  3 0LY*  3 0GA*  2 0LA
29 000 3                                                             *  1 0MC*  3 0SE
30 000  221B221B EST CANDIDATE AP MO       1 00000                    *  2 0ST
31 000  221C221C DET SYSTEM REQTS          2 0221A                    *  5 0ST* 10 0GO* 10 0SU
32 000  221D221D DEFINE SYSTEM DESIG       1 0221C                    *  5 0ST* 10 0GO* 10 0SU
33 000 1                                                             *  10 0KE* 10 0CO* 10 0SE
34 000 2                                                             *  10 0LY* 10 0GA* 10 0LA
35 000  221E221E FINALIZE BASELINES        1 0221D                    *  10 0ST* 10 0GO* 10 0SU
36 000 1                                                             *  3 0KE*  3 0CO*  3 0SE
37 000 2                                                             *  3 0LY*  3 0GA*  3 0LA
38 000  222A222A EST UAR EFFECTS REQ       1 0221E                    *  5 0ST* 10 0GO*  2 0SU
39 000  222B222B EST UAR EFFECTS SYS       2 0222A                    *  1 0ST*  6 0GO*  2 0SU
40 000 1                                                             *  5 0KE*  5 0CO*  5 0SE
41 000 2                                                             *  5 0LY*  5 0GA*  5 0LA
42 000  222C222C FINALIZE MODS DEFIN       2 02228                    *  3 0ST*  6 0GO*  2 0SU
43 000 1                                                             *  5 0KE*  5 0SE*  5 0LY
43 100 2                                                             *  5 0GA*  5 0LA
44 000  223A223A EST PLANT BASELINE        1 0 192                    *  4 0ST*  5 0SU
45 000  223B223B EST UAR EFFECTS REQ       2 0223A                    *  3 0ST* 10 0SU*  3 0LY
46 000 1                                                             *  3 0SE*  3 0LA
47 000  223C223C EST UAR EFFECTS SYS       2 0223B                    *  2 0ST*  5 0SU*  2 0KE
48 000 1                                                             *  3 0CO*  3 0SE*  3 0LY
48 100 2                                                             *  3 0GA*  3 0LA
49 000  223D223D FINALIZE PLANT MODS       1 0223C                    *  3 0ST*  7 0SU*  5 0KE
50 000 1                                                             *  6 0CO*  6 0SE*  6 0LY
50 100 2                                                             *  5 0GA*  5 0LA
51 000  224A224A SUMMARIZE COMPARISN       1 02223223C                *  6 0ST*  2 0GO*  2 0SU
51 100 1                                                             *  2 0MC
51 200  22432243 PREPARE REPORT            1 0224A222C223D            *  10 0ST* 10 0GO* 10 0SU
52 000 1                                                             *  8 0KE*  8 0CO*  6 0SE
53 000 1                                                             *  6 0LY*  6 0GA*  6 0LA
54 000 2
55 000  231A231A SELECT PLANTS             2 00000                    *  5 0IR*  2 0MC*  5 0JO
```

```
 56 000  230A232A SOU-PLANT EXPERIENC     1 00000          $  5 0IR$  1 0MC$  3 0JO
 57 000  230B232B PRCH CONT-PLANT EXP     2 0238A          $  5 0IR
 58 000  232C232C CONT-PLANT EXPERIEN     4 0838B831A      $  1 0IR+  4 0CB
 59 000  232D232D EVAL LIT OF PLANT       4 0231A          $  1 0IR$  1 0MC$  3 0CO
 60 000 1                                                 $  3 0SE$  3 0LY$  3 0QA
 60 100 2                                                 $  5 0JO
 61 000  232E232E PLANT VISITS/REPORT     3 0231A          $  3 0IR$  3 0MC$  5 0CO
 62 000 1                                                 $  5 0SE$  5 0LY$  5 0QA
 62 100 2                                                 + 60 0TB
 63 000  233A233A CONT-PLANT EXPERIEN     4 0231A8328      $  1 0IR+  6 0CB
 64 000  234A234A SURVEY MFGS             2 0238E          $  2 0IR$  5 0CO$  5 0SE
 65 000 1                                                 $  5 0LY$  5 0QA+ 15 0TB
 65 100 2                                                 $  5 0JO
 66 000  235A235A SUMMARIZE/COMPARE       4 0838E          $  3 0IR$  5 0CO$  5 0SE
 67 000 1                                                 $  5 0LY$  5 0QA$  5 0JO
 68 000  235B235B CONT-PLANT EXPERIEN     6 0838B          $  1 0IR+ 10 0CB
 69 000  236A236A RECOMMENDATIONS/REP     3 0234A          $  3 0IR$  1 0MC$  5 0CO
 70 000 1                                                 $  5 0SE$  5 0LY$  5 0QA
 70 100 2                                                 $  5 0JO
 71 000  510A510A TVA INTERFACE          17 00000          $  4 0IR+  4 0TB
 72 000  520A520A DOE INTERFACE          17 00000          $  4 0MI+ 16 0TB
 73 000  530A530A ACTIVITY PLANNING      17 00000          $  4 0MC+ 16 0TB
 74 000  0001END RAM                      0 0814C
 75 000  0002END PLANT DYN                1 0818B
 76 000  0003END AP DESIGN                5 0824B
 77 000  0004END EXPERIENCE               6 0836A
 78 000  0005END TASKS                    0 0000100030004
 79 000  0006END PROJECT                  0 0000551004E20A530A
 84 000 C
 85 000 TABLE R-
 86 000  CO COSTES
 87 000  DE DEATON
 88 000  GA GARRETT
 89 000  GO GOSS
 90 000  IR IRBY
 91 000  JO JOHNSTON
 93 000  KE KEARNS
 94 000  LA LARSEN
 95 000  LY LYNN
 96 000  MC MCCARTY
 97 000  MI MIDDLETON
 98 000  RA RELIABILITY/AVAILABILITY
 99 000  SE SEYMOUR
100 000  ST STONEMETZ
101 000  SU SUTHERLIN
102 000  CB CONTRACT B.%OF TOTAL
103 000  TB TRAVEL B.% OF TOTAL
104 000 C
105 000 EOF
```

'PLATEN 140

'SET F 5 DP/ND TA

'START EXECUTE

MAIN MENU

SELECT ONE OF THE FOLLOWING

1 - DEFINE AND LOAD NETWORK SOURCE FILE
2 - EDIT NETWORK
3 - PROCESS NETWORK
4 - SET/RESET AUTO-HARDCOPY(CURRENTLY OFF)
5 - SET/RESET SEGMENTATION OF PLOTS/TABLES(CURRENTLY OFF)
6 - DISPLAY INPUT ERROR REPORT
7 - DISPLAY NETWORK ECHO REPORT
8 - DISPLAY RESOURCE ECHO REPORT
9 - DISPLAY PREDECESSOR/SUCCESSOR REPORT
10 - DISPLAY ACTIVITY SCHEDULING REPORT
11 - DISPLAY RESOURCE HISTOGRAM TABLES
12 - DISPLAY FULL TITLE TABLE
13 - DISPLAY INPUT-ORDER TIMELINE PLOTS
14 - DISPLAY WATERFALL ORDER TIMELINE PLOTS
15 - DISPLAY RESOURCE HISTOGRAM PLOTS
16 - DISPLAY ALL REPORTS AND PLOTS
17 - DISPLAY ALL REPORTS
18 - DISPLAY ALL PLOTS
0 - TERMINATE

?01

```
 CHOOSE ONE OF THE FOLLOWING
   1 - DATA READ FROM SIGMA V BUILT FILE
   2 - DATA READ FROM IGDS BUILT FILE
?2
 WHAT IS THE TITLE OF THIS RUN-UP TO 36CHAR
?TEST DATA
 UNITS TO BE USED
?1 (METRIC)
```

```
                              MAIN MENU

SELECT ONE OF THE FOLLOWING

     1 - DEFINE AND LOAD NETWORK SOURCE FILE
     2 - EDIT NETWORK
     3 - PROCESS NETWORK
     4 - SET/RESET AUTO-HARDCOPY(CURRENTLY OFF )
     5 - SET/RESET SEGMENTATION OF PLOTS/TABLES(CURRENTLY OFF )
     6 - DISPLAY INPUT ERROR REPORT
     7 - DISPLAY NETWORK ECHO REPORT
     8 - DISPLAY RESOURCE ECHO REPORT
     9 - DISPLAY PREDECESSOR/SUCCESSOR REPORT
    10 - DISPLAY ACTIVITY SCHEDULING REPORT
    11 - DISPLAY RESOURCE HISTOGRAM TABLES
    12 - DISPLAY FULL TITLE TABLE
    13 - DISPLAY INPUT-ORDER TIMELINE PLOTS
    14 - DISPLAY WATERFALL ORDER TIMELINE PLOTS
    15 - DISPLAY RESOURCE HISTOGRAM PLOTS
    16 - DISPLAY ALL REPORTS AND PLOTS
    17 - DISPLAY ALL REPORTS
    18 - DISPLAY ALL PLOTS
     0 - TERMINATE

?03
```

MAIN MENU

SELECT ONE OF THE FOLLOWING

1 - DEFINE AND LOAD NETWORK SOURCE FILE
2 - EDIT NETWORK
3 - PROCESS NETWORK
4 - SET/RESET AUTO-HARDCOPY(CURRENTLY OFF)
5 - SET/RESET SEGMENTATION OF PLOTS/TABLES(CURRENTLY OFF)
6 - DISPLAY INPUT ERROR REPORT
7 - DISPLAY NETWORK ECHO REPORT
8 - DISPLAY RESOURCE ECHO REPORT
9 - DISPLAY PREDECESSOR/SUCCESSOR REPORT
10 - DISPLAY ACTIVITY SCHEDULING REPORT
11 - DISPLAY RESOURCE HISTOGRAM TABLES
12 - DISPLAY FULL TITLE TABLE
13 - DISPLAY INPUT-ORDER TIMELINE PLOTS
14 - DISPLAY WATERFALL ORDER TIMELINE PLOTS
15 - DISPLAY RESOURCE HISTOGRAM PLOTS
16 - DISPLAY ALL REPORTS AND PLOTS
17 - DISPLAY ALL REPORTS
18 - DISPLAY ALL PLOTS
0 - TERMINATE

?04

```
 THE AUTOMATIC HARDCOPY IS CURRENTLY OFF
 INPUT A 1 IF YOU WISH IT ON OR A 2 IF YOU WISH IT OFF
?1
```

MAIN MENU

SELECT ONE OF THE FOLLOWING

1 - DEFINE AND LOAD NETWORK SOURCE FILE
2 - EDIT NETWORK
3 - PROCESS NETWORK
4 - SET RESET AUTO-HARDCOPY(CURRENTLY ON)
5 - SET RESET SEGMENTATION OF PLOTS/TABLES(CURRENTLY OFF)
6 - DISPLAY INPUT ERROR REPORT
7 - DISPLAY NETWORK ECHO REPORT
8 - DISPLAY RESOURCE ECHO REPORT
9 - DISPLAY PREDECESSOR/SUCCESSOR REPORT
10 - DISPLAY ACTIVITY SCHEDULING REPORT
11 - DISPLAY RESOURCE HISTOGRAM TABLES
12 - DISPLAY FULL TITLE TABLE
13 - DISPLAY INPUT-ORDER TIMELINE PLOTS
14 - DISPLAY WATERFALL ORDER TIMELINE PLOTS
15 - DISPLAY RESOURCE HISTOGRAM PLOTS
16 - DISPLAY ALL REPORTS AND PLOTS
17 - DISPLAY ALL REPORTS
18 - DISPLAY ALL PLOTS
0 - TERMINATE

?16

TEST DATA
SCHEDULING PROGRAM INPUT ECHO REPORT

SEQ NBR	TASK LABEL	TASK TITLE	DUR	NO PRED	PRED 1	PRED 2	PRED 3	PRED 4	PRED 5	NO S/R	S/R 1	S/R 2	S/R 3	S/R 4	S/R 5
1	0000	START	0	0						0					
2	191	191 AWARD A/E CONTRACT	4	1	0000					0					
3	192	192 PLANT CONCEPT ENG PK	2	1	191					0					
4	193	193 60% PROCESS PLANT PK	6	1	192					0					
5	211A	211A COMPONENTS LIST	1	1	0000					2 R	8	0LA	R 5	0JO	
6	211B	211B SOW-COMP DATA COLLE	1	1	213A					2 R	8	0LA	R 1	0MI	
7	212A	212A PRCH CONT-COMP D CL	2	1	211B					1 R	5	0LA			
8	212B	212B CONTRACT-COMP D COL	6	1	212A					2 R	3	0LA	+ 17	0CB	
9	213A	213A RAM METHODS REVIEW	1	1	0000					2 R	8	0RA	R 1	0MI	
10	213B	213B SOW-RAM MODEL	1	1	213A					1 R	8	0RA			
11	213C	213C PRCH CONT-RAM MODEL	2	1	213B					1 R	5	0RA			
12	213D	213D CONTRACT-RAM MOD DU	4	1	213C					2 R	1	0RA	+ 19	0CB	
13	214A	214A INITIAL RAM ASSESSM	4	2	213D	192				3 R	7	0RA	R 3	0MI	R 5 0JO
14	214B	214B 60% DESIGN QUAL ASM	2	2	214A	193				3 R	8	0RA	R 2	0MI	R 5 0JO
15	214C	214C 60% DESIGN QUANT AS	4	2	214A	193				3 R	8	0RA	R 3	0MI	R 5 0JO
16	214D	214D CONTRACT-RAM MOD AS	11	1	213D					2 R	1	0RA	+ 13	0CB	
17	215A	215A DYNAMICS MODEL DEFI	1	1	0000					2 R	8	0DE	R 2	0MI	
18	215B	215B SOW-DYNAMICS MODEL	1	1	215A					1 R	8	0DE			
19	215C	215C PRCH CONT-DYNAM MOD	2	1	215B					1 R	5	0DE			
20	215D	215D CONT-DYN MODEL DEVE	4	1	215C					2 R	1	0DE	+ 14	0CB	
21	216A	216A INITIAL DYNAMICS AS	4	2	215D	192				2 R	8	0DE	R 2	0MI	
22	216B	216B 60% DESIGN ASSESSME	6	2	215A	193				2 R	8	0DE	R 3	0MI	
23	216C	216C CONT-DYN MODEL ASSE	14	1	215D					2 R	1	0DE	+ 8	0CB	
24	216D	216D DESIGN RECOMMENDATI	8	1	216B					2 R	10	0DE	R 5	0MI	
25	221A	221A COMPLETE DESIN TOOL	2	1	0000					11 +	6	0CB	R 7	0ST	R 10 0QO R 10 0SU R 3 0KE
26	221B	221B EST CANDIDATE AP MO	1	1	0000					1 R	2	0ST			
27	221C	221C DET SYSTEM REQTS	2	1	221A					3 R	5	0ST	R 10	0QO	R 10 0SU
28	221D	221D DEFINE SYSTEM DESIG	1	1	221C					9 R	5	0ST	R 10	0QO	R 10 0SU R 10 0KE R 10 0CO
29	221E	221E FINALIZE BASELINES	1	1	221D					9 R	10	0ST	R 10	0QO	R 10 0SU R 3 0KE R 3 0CO
30	222A	222A EST VAR EFFECTS REQ	1	1	221E					3 R	5	0ST	R 10	0QO	R 2 0SU
31	222B	222B EST VAR EFFECTS SYS	2	1	222A					9 R	1	0ST	R 6	0QO	R 2 0SU R 5 0KE R 5 0CO
32	222C	222C FINALIZE MODS DEFIN	2	1	222B					8 R	3	0ST	R 6	0QO	R 2 0SU R 5 0KE R 5 0SE
33	223A	223A EST PLANT BASELINE	1	1	192					2 R	4	0ST	R 5	0SU	
34	223B	223B EST VAR EFFECTS REQ	2	1	223A					6 R	3	0ST	R 10	0SU	R 3 0LV R 3 0SE R 3 0LA
35	223C	223C EST VAR EFFECTS SYS	2	1	223B					8 R	2	0ST	R 5	0SU	R 2 0KE R 3 0CO R 3 0SE
36	223D	223D FINALIZE PLANT MODS	1	1	223C					8 R	3	0ST	R 7	0SU	R 5 0KE R 6 0CO R 5 0SE
37	224A	224A SUMMARIZE COMPARISN	1	2	222B	223C				4 R	5	0ST	R 2	0QO	R 2 0SU R 2 0MC
38	224B	224B PREPARE REPORT	1	3	224A	222C	223D			9 R	10	0ST	R 10	0QO	R 10 0SU R 6 0KE R 6 0CO
39	231A	231A SELECT PLANTS	2	1	0000					3 R	5	0IR	R 2	0MC	R 5 0JO
40	232A	232A SOW-PLANT EXPERIENC	1	1	0000					3 R	5	0IR	R 1	0MC	R 3 0JO
41	232B	232B PRCH CONT-PLANT EXP	2	1	232A					1 R	5	0IR			
42	232C	232C CONT-PLANT EXPERIEN	4	2	232B	231A				2 R	1	0IR	+ 4	0CB	
43	232D	232D EVAL LIT OF PLANT	4	1	231A					7 R	1	0IR	R 1	0MC	R 3 0CO R 3 0SE R 3 0LV
44	232E	232E PLANT VISITS/REPORT	3	1	231A					7 R	3	0IR	R 2	0MC	R 5 0CO R 5 0SE R 5 0LV
45	233A	233A CONT-PLANT EXPERIEN	4	2	231A	232B				2 R	1	0IR	+ 6	0CB	
46	234A	234A SURVEY REQS	2	1	232E					7 R	2	0IR	R 5	0CO	R 5 0SE R 5 0LV R 5 0QA
47	235A	235A SUMMARIZE/COMPARE	4	1	232E					6 R	3	0IR	R 5	0CO	R 5 0SE R 5 0LV R 5 0QA
48	235B	235B CONT-PLANT EXPERIEN	6	1	232B					2 R	1	0IR	+ 10	0CB	
49	236A	236A RECOMMENDATIONS/REP	3	1	234A					7 R	3	0IR	R 1	0MC	R 5 0CO R 5 0SE R 5 0LV
50	510A	510A TUA INTERFACE	17	1	0000					2 R	4	0IR	+ 4	0TB	

51	520A	520A DOE INTERFACE	17	1	0000	2 x 4.0FI + 16 0T8	
52	530A	530A ACTIVITY PLANNING	17	1	0000	2 x 4.0FC + 16 0T6	
53	0001	END RAP	0	1	214C	0	
54	0002	END PLANT DYN	1	1	216D	0	
55	0003	END AP DESIGN	5	1	224B	0	
56	0004	END EXPERIENCE	6	1	238A	0	
57	0005	END TASKS	0	3	0001 0003 0004	0	
58	0006	END PROJECT	0	4	0005 510A 520A 530A	0	

RESOURCE TABLE

RSC CODE	DESCRIPTION
CO	COSTES
DE	DEATON
GA	GARRETT
GO	GOSS
IR	IRBY
JO	JOHNSTON
KE	KEARNS
LA	LARSEN
LY	LYNN
MC	MCCARTY
MI	MIDDLETON
RA	RELIABILITY/AVAILABILITY
SE	SEYMOUR
ST	STONEMETZ
SU	SUTHERLIN
C$	CONTRACT $ %OF TOTAL
T$	TRAVEL $ % OF TOTAL

TEST DATA
PREDECESSOR-SUCCESSOR TABLE

SEQ NBR	TASK LABEL	TASK TITLE	NO PRED	PRED 1	PRED 2	PRED 3	PRED 4	PRED 5	PRED 6	PRED 7	NO SUCR	SUCR 1	SUCR 2	SUCR 3	SUCR 4	SUCR 5	SUCR 6	SUCR 7
1	0000	START	0								11	191	211A	213A	215A	221A	221B	231A
2	191	191 AWARD A/E CONTRACT	1	0000							1	192						
3	192	192 PLANT CONCEPT ENG PK	1	191							4	193	214A	216A	223A			
4	193	193 60% PROCESS PLANT PK	1	192							3	214B	214C	216B				
5	211A	211A COMPONENTS LIST	1	0000							0							
6	211B	211B SOW-COMP DATA COLLE	1	213A							1	212A						
7	212A	PRCH CONT-COMP D CL	1	211B							1	212B						
8	212B	212B CONTRACT-COMP D COL	1	212A							0							
9	213A	213A RAM METHODS REVIEW	1	0000							2	211B	213B					
10	213B	213B SOW-RAM MODEL	1	213A							1	213C						
11	213C	PRCH CONT-RAM MODEL	1	213B							1	213D						
12	213D	213D CONTRACT-RAM MOD DV	1	213C							2	214A	214D					
13	214A	214A INITIAL RAM ASSESSM	2	213D	192						2	214B	214C					
14	214B	214B 60% DESIGN QUAL ASM	2	214A	193						0							
15	214C	214C 60% DESIGN QUANT AS	2	214A	193						1	0001						
16	214D	214D CONTRACT-RAM MOD AS	1	213D							0							
17	215A	215A DYNAMICS MODEL DEFI	1	0000							1	215B						
18	215B	215B SOW-DYNAMICS MODEL	1	215A							1	215C						
19	215C	215C PRCH CONT-DYNAM MOD	1	215B							1	215D						
20	215D	215D CONT-DYN MODEL DEVE	1	215C							2	216A	216C					
21	216A	216A INITIAL DYNAMICS AS	2	215D	192						1	216B						
22	216B	216B 60% DESIGN ASSESSME	2	216A	193						1	216D						
23	216C	216C CONT-DYN MODEL ASSE	1	215D							0							
24	216D	216D DESIGN RECOMMENDATI	1	216B							1	0002						
25	221A	221A COMPLETE DESIN TOOL	1	0000							1	221C						
26	221B	221B EST CANDIDATE AP MO	1	0000							0							
27	221C	221C DET SYSTEM REQTS	1	221A							1	221D						
28	221D	221D DEFINE SYSTEM DESIG	1	221C							1	221E						
29	221E	221E FINALIZE BASELINES	1	221D							1	222A						
30	222A	222A EST VAR EFFECTS REQ	1	221E							1	222B						
31	222B	222B EST VAR EFFECTS SYS	1	222A							2	222C	224A					
32	222C	222C FINALIZE MODS DEFIN	1	222B							1	224B						
33	223A	223A EST PLANT BASELINE	1	192							1	223B						
34	223B	223B EST VAR EFFECTS REQ	1	223A							1	223C						
35	223C	223C EST VAR EFFECTS SYS	1	223B							2	223D	224A					
36	223D	223D FINALIZE PLANT MODS	1	223C							1	224B						
37	224A	224A SUMMARIZE COMPARISN	2	222B	223C						1	224B						
38	224B	224B PREPARE REPORT	3	224A	222C	223D					1	0003						
39	231A	231A SELECT PLANTS	1	0000							4	232C	232D	232E	233A			
40	232A	232A SOW-PLANT EXPERIENC	1	0000							1	232B						
41	232B	232B PRCH CONT-PLANT EXP	1	232A							3	232C	233A	235B				
42	232C	232C CONT-PLANT EXPERIEN	2	232B	231A						0							
43	232D	232D EVAL LIT OF PLANT	1	231A							0							
44	232E	232E PLANT VISIT. /REPORT	1	231A							2	234A	236A					
45	233A	233A CONT-PLANT E..PERIEN	2	231A	232B						0							
46	234A	234A SURVEY MFGS	1	232E							1	236A						
47	235A	235A SUMMARIZE/COMPARE	1	232E							0							
48	235B	235B CONT-PLANT EXPERIEN	1	232B							0							
49	236A	236A RECOMMENDATIONS/REP	1	234A							1	0004						
50	510A	510A TVA INTERFACE	1	0000							1	0006						

TEST DATA
PREDECESSOR-SUCCESSOR TABLE

SEQ NBR	TASK LABEL	TASK TITLE	NO PRED	PRED 1	PRED 2	PRED 3	PRED 4	PRED 5	PRED 6	PRED 7	NO SUCR	SUCR 1	SUCR 2	SUCR 3	SUCR 4	SUCR 5	SUCR 6	SUCR 7
51	520A	520A DOE INTERFACE	1	0000							1	0006						
52	530A	530A ACTIVITY PLANNING	1	0000							1	0006						
53	0001	END RAM	1	214C							1	0005						
54	0002	END PLANT DYN	1	216D							0							
55	0003	END AP DESIGN	1	224B							1	0005						
56	0004	END EXPERIENCE	1	236A							1	0005						
57	0005	END TASKS	3	0001	0003	0004					1	0006						
58	0006	END PROJECT	4	0005	510A	520A	530A				0							

TEST DATA
TASK SCHEDULING TABLE

SEQ NBR	TASK LABEL	TASK TITLE	DUR	SHIFT	SCALED DUR	SLACK TIME	START TIME	STOP TIME
1	0000	START	0	3	0	0	0	0
3	213A	213A RAM METHODS REVIEW	1	3	1 0	0	0	1 0
17	215A	215A DYNAMICS MODEL DEFI	1	3	1 0	0	0	1 0
25	221A	221A EST CANDIDATE AP MO	1	3	1 0	0	0	1 0
40	232A	232A SOW-PLANT EXPERIENC	1	3	1 0	0	0	1 0
5	211A	211A COMPONENTS LIST	1	3	1 1	0	0	1 1
25	221A	221A COMPLETE DESIN TOOL	2	3	2 0	0	0	2 0
39	231A	231A SELECT PLANTS	2	3	2 0	0	0	2 0
2	191	191 AWARD A/E CONTRACT	4	3	3 5	0	0	3 5
50	510A	510A TVA INTERFACE	17	3	17 0	0	0	17 0
51	520A	520A DOE INTERFACE	17	3	17 0	0	0	17 0
52	530A	530A ACTIVITY PLANNING	17	3	17 0	0	0	17 0
6	211B	211B SOW-COMP DATA COLLE	1	3	1 0	0	1 0	2 0
10	213B	213B SOW-RAM MODEL	1	3	1 0	0	1 0	2 0
18	215B	215B SOW-DYNAMICS MODEL	1	3	1 0	0	1 0	2 0
41	232B	232B PRCH CONT-PLANT EXP	2	3	2 0	0	1 0	3 0
7	212A	212A PRCK CONT-COMP D CL	2	3	2 0	0	2 0	4 0
11	213C	213C PRCH CONT-RAM MODEL	2	3	2 0	0	2 0	4 0
19	215C	215C PRCH CONT-DYNAM MOD	2	3	2 0	0	2 0	4 0
27	221C	221C DET SYSTEM REQTS	2	3	2 0	0	2 0	4 0
44	232E	232E PLANT VISITS/REPORT	3	3	3 0	0	2 0	5 0
43	232D	232D EVAL LIT OF PLANT	4	3	4 0	0	2 0	6 0
42	232C	232C CONT-PLANT EXPERIEN	4	3	4 0	0	3 0	7 0
45	233A	233A CONT-PLANT EXPERIEN	4	3	4 0	0	3 0	7 0
48	235B	235B CONT-PLANT EXPERIEN	6	3	6 0	0	3 0	9 0
3	192	192 PLANT CONCEPT ENG PK	2	3	2 0	0	3 5	5 5
28	221D	221D DEFINE SYSTEM DESIG	1	3	1 0	0	4 0	5 0
12	213D	213D CONTRACT-RAM MOD DU	4	3	4 0	0	4 0	8 0
20	215D	215D CONT-DYN MODEL DEVE	4	3	4 0	0	4 0	8 0
8	212B	212B CONTRACT-COMP D COL	6	3	6 0	0	4 0	10 0
29	221E	221E FINALIZE BASELINES	1	3	1 0	0	5 0	6 0
46	234A	234A SURVEY MFGS	2	3	2 0	0	5 0	7 0
47	235A	235A SUMMARIZE/COMPARE	4	3	4 0	0	5 0	9 0
33	223A	223A EST PLANT BASELINE	1	3	1 0	0	5 5	6 5
4	193	193 60% PROCESS PLANT PK	6	3	6 0	0	5 5	11 5
30	222A	222A EST VAR EFFECTS REQ	1	3	1 0	0	6 0	7 0
34	223B	223B EST VAR EFFECTS REQ	2	3	2 0	0	6 5	8 5
31	222B	222B EST VAR EFFECTS SYS	2	3	2 0	0	7 0	9 0
49	236A	236A RECOMMENDATIONS/REP	3	3	3 0	0	7 0	10 0
13	214A	214A INITIAL RAM ASSESSM	4	3	4 0	0	8 0	12 0
21	216A	216A INITIAL DYNAMICS AS	4	3	4 0	0	8 0	12 0
16	214D	214D CONTRACT-RAM MOD AS	11	3	11 0	0	8 0	19 0
23	216C	216C CONT-DYN MODEL ASSE	14	3	14 0	0	8 0	22 0
35	223C	223C EST VAR EFFECTS SYS	2	3	2 0	0	8 5	10 5
32	222C	222C FINALIZE MODS DEFIN	2	3	2 0	0	9 0	11 0
56	0004	END EXPERIENCE	6	3	6 0	0	10 0	16 0
36	223D	223D FINALIZE PLANT MODS	1	3	1 0	0	10 5	11 5
37	224A	224A SUMMARIZE COMPARISN	1	3	1 0	0	10 5	11 5
38	224B	224B PREPARE REPORT	1	3	1 0	0	11 5	12 5
14	214B	214B 60% DESIGN QUAL ASM	2	3	1 5	0	12 0	13 5

TEST DATA
TASK SCHEDULING TABLE

SEQ NBR	TASK LABEL	TASK TITLE	DUR	SHIFT	SCALED DUR	SLACK TIME	START TIME	STOP TIME
15	214C	214C 60% DESIGN QUANT AS	4	3	4.0	0	12.0	16.0
22	216E	216B 60% DESIGN ASSESSME	6	3	6.0	0	12.0	18.0
55	0003	END AP DESIGN	5	3	5.0	0	12.5	17.5
53	0001	END RAM	0	3	0	0	16.0	16.0
57	0005	END TASKS	0	3	0	0	17.5	17.5
58	0006	END PROJECT	0	3	0	0	17.5	17.5
24	216D	216D DESIGN RECOMMENDATI	2	3	2.0	0	18.0	20.0
54	0002	END PLANT DYN	1	3	1.0	0	20.0	21.0

RESOURCE HISTOGRAM FOR COSTES
TIME UNITS MONTHS MAXIMUM LEVEL 18
INITIAL VALUE 0 CONSTRAINED RESOURCE OFF

TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL
0	3	2 0	8	4 5	18	5 0	16	6 0	10	7 0	15
6 5	18	9 0	8	10 0	3	10 5	6	12 5	0		

RESOURCE HISTOGRAM FOR DEATON
TIME UNITS MONTHS MAXIMUM LEVEL 11
INITIAL VALUE 0 CONSTRAINED RESOURCE OFF

TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL
0	8	2 0	5	4 0	1	8 0	9	18 0	11	20 0	1
22 0	0										

RESOURCE HISTOGRAM FOR GARRETT
TIME UNITS MONTHS MAXIMUM LEVEL 18
INITIAL VALUE 0 CONSTRAINED RESOURCE OFF

TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL
0	3	2 0	8	4 0	18	5 0	16	6 0	10	7 0	15
8 5	18	9 0	13	10 0	8	10 5	10	11 0	5	11 5	6
12 5	0										

RESOURCE HISTOGRAM FOR GOSS
TIME UNITS MONTHS MAXIMUM LEVEL 10
INITIAL VALUE 0 CONSTRAINED RESOURCE OFF

TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL
0	10	7 0	6	10 5	8	11 0	2	11 5	10	12 5	0

RESOURCE HISTOGRAM FOR IRBY
TIME UNITS MONTHS MAXIMUM LEVEL 14
INITIAL VALUE 0 CONSTRAINED RESOURCE OFF

TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL
0	14	2 0	13	3 0	11	5 0	13	6 0	12	7 0	11
9 0	7	10 0	4	17 0	0						

RESOURCE HISTOGRAM FOR JOHNSTON
TIME UNITS MONTHS MAXIMUM LEVEL 15
INITIAL VALUE 0 CONSTRAINED RESOURCE OFF

TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL

0	13	1 0	10	1 1	5	5 0	15	6 0	10	8 0	16				
9 0	10	10 0	5	12 0	10	13 5	5	16 0	0						

RESOURCE HISTOGRAM FOR KEARNS
TIME UNITS MONTHS MAXIMUM LEVEL 10
INITIAL VALUE 0 CONSTRAINED RESOURCE OFF

TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL
0	3	2 0	0	4 0	10	5 0	3	6 0	0	7 0	5		
4 5	7	10 5	10	11 0	5	11 5	6	18 5	0				

RESOURCE HISTOGRAM FOR LARSEN
TIME UNITS MONTHS MAXIMUM LEVEL 18
INITIAL VALUE 0 CONSTRAINED RESOURCE OFF

TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL
0	10	1 0	18	1 1	10	2 0	5	4 0	13	6 0	6		
6 0	3	6 5	6	7 0	11	10 0	8	10 5	10	11 0	5		
11 5	6	12 5	0										

RESOURCE HISTOGRAM FOR LYNN
TIME UNITS MONTHS MAXIMUM LEVEL 18
INITIAL VALUE 0 CONSTRAINED RESOURCE OFF

TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL
0	3	2 0	8	4 0	18	5 0	16	6 0	10	6 5	13		
7 0	18	9 0	13	10 0	8	10 5	10	11 0	5	11 5	6		
12 5	0												

RESOURCE HISTOGRAM FOR MCCARTY
TIME UNITS MONTHS MAXIMUM LEVEL 8
INITIAL VALUE 0 CONSTRAINED RESOURCE OFF

TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL
0	8	1 0	7	2 0	8	5 0	5	6 0	4	7 0	5		
10 0	4	10 5	6	11 5	4	17 0	0						

RESOURCE HISTOGRAM FOR MIDDLETON
TIME UNITS MONTHS MAXIMUM LEVEL 12
INITIAL VALUE 0 CONSTRAINED RESOURCE OFF

TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL
0	7	1 0	5	2 0	4	8 0	9	12 0	12	13 5	10		
16 0	7	17 0	3	18 0	5	20 0	0						

RESOURCE HISTOGRAM FOR RELIABILITY/AVAILABILITY
TIME UNITS MONTHS MAXIMUM LEVEL 17
INITIAL VALUE 0 CONSTRAINED RESOURCE OFF

TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL

0	8	8 0	8	4 0	1	8 0	8	12 0	17	13 5	9
16 0	1	19 0	0								

RESOURCE HISTOGRAM FOR SEYMOUR
TIME UNITS MONTHS MAXIMUM LEVEL 18
INITIAL VALUE 0 CONSTRAINED RESOURCE OFF

TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL
0	3	2 0	8	4 0	18	5 0	16	6 0	10	6 5	13
9 0	15	9 0	13	10 0	8	10 5	10	11 0	6	11 5	8
12 5	0										

RESOURCE HISTOGRAM FOR STONEMETZ
TIME UNITS MONTHS MAXIMUM LEVEL 14
INITIAL VALUE 0 CONSTRAINED RESOURCE OFF

TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL
0	9	1 0	7	2 0	5	5 0	10	5 5	14	6 0	9
6	8	7 0	4	8 5	3	9 0	5	10 5	11	11 0	8
11	10	12 5	0								

RESOURCE HISTOGRAM FOR SUTHERLIN
TIME UNITS MONTHS MAXIMUM LEVEL 15
INITIAL VALUE 0 CONSTRAINED RESOURCE OFF

TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL
0	10	5 5	15	8 0	7	6 5	12	8 5	7	10 5	11
11 0	9	11 5	10	12 5	0						

RESOURCE HISTOGRAM FOR CONTRACT $.%OF TOTAL
TIME UNITS MONTHS MAXIMUM LEVEL 97
INITIAL VALUE 0 CONSTRAINED RESOURCE OFF

TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL
0	6	3 0	26	4 0	76	8 0	97				

RESOURCE HISTOGRAM FOR TRAVEL $.% OF TOTAL
TIME UNITS MONTHS MAXIMUM LEVEL 111
INITIAL VALUE 0 CONSTRAINED RESOURCE OFF

TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL	TIME	LEVEL
0	36	2 0	96	6 0	111						

NETWORK DESCRIPTION

TEST DATA

TASK LABEL	TASK DESCRIPTION	DURATION (MONTHS)	START TIME	STOP TIME
000A	START	0	0	0
	NO PREDECESSORS			
	SUCCESSORS			
	191 AWARD A/E CONTRACT 211A COMPONENTS LIST 213A RAM METHODS REVIEW 215A DYNAMICS MODEL DEFI 221A COMPLETE DESIN TOOL 221B EST CANDIDATE AP MO 231A SELECT PLANTS 232A SOU-PLANT EXPERIENC 510A TVA INTERFACE 520A DOE INTERFACE 530A ACTIVITY PLANNING			
213A	213A RA METHODS REVIEW	1 0	0	1 0
	PREDECESSORS			
	START			
	SUCCESSORS			
	211B SOU-COMP DATA COLLE 213B SOU-RAM MODEL			
215A	215A DYNAMICS MODEL DEFI	1 0	0	1 0
	PREDECESSORS			
	START			
	SUCCESSORS			
	215B SOU-DYNAMICS MODEL			
221B	221B EST CANDIDATE AP MO	1 0	0	1 0
	PREDECESSORS			
	START			
	NO SUCCESSORS			
232A	232A SOU-PLANT EXPERIENC	1 0	0	1 0

PREDECESSORS

START

SUCCESSORS

236B PROM CONT-PLANT EXP

NETWORK DESCRIPTION

TEST DATA

TASK NUMBER	TASK DESCRIPTION	DURATION (MONTHS)	START TIME	EARLY START
211A	211A COMPONENTS LIST	1 1	0	1 1
	PREDECESSORS			
	START			
	NO SUCCESSORS			
221A	221A COMPLETE DESIN 'OCL	2 0	0	2 0
	PREDECESSORS			
	START			
	SUCCESSORS			
	221C DET SYSTEM REQTS			
231A	231A SELECT PLANTS	2 0	0	2 0
	PREDECESSORS			
	START			
	SUCCESSORS			
	23.C CONT-PLANT EXPERIEN 232D EVAL LIT OF PLANT 232E PLANT VISITS/REPORT 233A CONT-PLANT EXPERIEN			
191	191 AWARD A/E CONTRACT	3 5	0	3 5
	PREDECESSORS			
	START			
	SUCCESSORS			
	192 PLANT CONCEPT ENG PK			
518A	518A TVA INTERFACE	17 0	0	17 0
	PREDECESSORS			
	START			
	SUCCESSORS			
	END PROJECT			
520A	520A DOE INTERFACE	17 0	0	
	PREDECESSORS			
	START			
	SUCCESSORS			
	END PROJECT			

TASK LABEL	TASK DESCRIPTION	DURATION (MONTHS)	START TIME	STOP TIME
530A	530A ACTIVITY PLANNING	17.0	0	17.0
	PREDECESSORS			
	START			
	SUCCESSORS			
	END PROJECT			
211B	211B SOW-COMP DATA COLLE	1.0	1.0	2.0
	PREDECESSORS			
	213A RAM METHODS REVIEW			
	SUCCESSORS			
	212A PRCH CONT-COMP D CL			
213B	213B SOW-RAM MODEL	1.0	1.0	2.0
	PREDECESSORS			
	213A RAM METHODS REVIEW			
	SUCCESSORS			
	213C PRCH CONT-RAM MODEL			
215B	215B SOW-DYNAMICS MODEL	1.0	1.0	2.0
	PREDECESSORS			
	215A DYNAMICS MODEL DEFI			
	SUCCESSORS			
	215C PRCH CONT-DYNAM MOD			
232B	232B PRCH CONT-PLANT EXP	2.0	1.0	3.0
	PREDECESSORS			
	232A SOW-PLANT EXPERIEN			

SUCCESSORS

232C CONT-PLANT EXPERIEN
233A CONT-PLANT EXPERIEN
235A CONT-PLANT EXPERIEN

212A	PRCK CONT-COMP D CL	2.0	2.0

PREDECESSORS

211B SOW-COMP DATA COLLE

SUCCESSORS

212B CONTRACT-COMP D COL

TASK LABEL	TASK DESCRIPTION	DURATION (MONTHS)	START TIME	STOP TIME
213C	213C PRCH CONT-RAM MODEL	2 0	2 0	4 0
	PREDECESSORS			
	213B SOU-RAM MODEL			
	SUCCESSORS			
	213D CONTRACT-RAM MOD DU			
215C	215C PRCH CONT-DYNAM MOD	2 0	2 0	4 0
	PREDECESSORS			
	215B SOU-DYNAMICS MODEL			
	SUCCESSORS			
	215D CONT-DYN MODEL DEVE			
221C	221C DET SYSTEM REQTS	2 0	2 0	4 0
	PREDECESSORS			
	221A COMPLETE DESIN TOOL			
	SUCCESSORS			
	221D DEFINE SYSTEM DESIG			
232E	232E PLANT VISITS/REPORT	3 0	2 0	5 0
	PREDECESSORS			
	231A SELECT PLANTS			
	SUCCESSORS			
	234A SURVEY RFQS 236A SUMMARIZE/COMPARE			
232D	232D EVAL LIT OF PLANT	4 0	2 0	6 0
	PREDECESSORS			
	231A SELECT PLANTS			
	NO SUCCESSORS			
232C	232C CONT-PLANT EXPENIEN	4 0	3 0	
	PREDECESSORS			
	232B PRCH CONT-PLANT EXP 231A SELECT PLANTS			
	NO SUCCESSORS			

NETWORK DESCRIPTION

TEST DATA

TASK LABEL	TASK DESCRIPTION	DURATION (MONTHS)	START TIME	STOP TIME
231A	231A CONT-PLANT EXPERIEN	4 0	3 0	7 0
	PREDECESSORS			
	231A SELECT PLANTS 232B PRCH CONT-PLANT EXP			
	NO SUCCESSORS			
235B	235B CONT-PLANT EXPERIEN	6 0	3 0	9 0
	PREDECESSORS			
	232B PRCH CONT-PLANT EXP			
	NO SUCCESSORS			
192	192 PLANT CONCEPT ENG PK	2 0	3 5	5 5
	PREDECESSORS			
	191 AWARD A/E CONTRACT			
	SUCCESSORS			
	193 60% PROCESS PLANT PK 214A INITIAL RAM ASSESSM 216A INITIAL DYNAMICS AS 223A EST PLANT BASELINE			
221D	221D DEFINE SYSTEM DESIG	1 0	4 0	5 0
	PREDECESSORS			
	221C DET SYSTEM REQTS			
	SUCCESSORS			
	221E FINALIZE BASELINES			
213D	213D CONTRACT-RAM MOD DU	4 0	4 0	8 0
	PREDECESSORS			
	213C PRCH CONT-RAM MODEL			
	SUCCESSORS			

TASK LABEL	TASK DESCRIPTION	DURATION (MONTHS)	START TIME	STOP TIME
215D	215D CONT-DYN MODEL DEVE	4 0	4 0	8 0
	PREDECESSORS			
	215C PRCH CONT-DYNAM MOD			
	SUCCESSORS			
	216A INITIAL DYNAMICS AS 216C CONT-DYN MODEL ASSE			
212B	212B CONTRACT-COMP D COL	6 0	4 0	10 0
	PREDECESSORS			
	212A PRCK CONT-COMP D CL			
	NO SUCCESSORS			
221E	221E FINALIZE BASELINES	1 0	5 0	6 0
	PREDECESSORS			
	221D DEFINE SYSTEM DESIG			
	SUCCESSORS			
	222A EST VAR EFFECTS REQ			
234A	234A SURVEY MFGS	2 0	5 0	7 0
	PREDECESSORS			
	232E PLANT VISITS/REPORT			
	SUCCESSORS			
	236A RECOMMENDATIONS/REP			
235A	235A SUMMARIZE/COMPARE	4 0	5 0	9 0
	PREDECESSORS			
	232E PLANT VISITS/REPORT			
	NO SUCCESSORS			

223A
6 5

823A EST PLANT BASELINE 1 0 5 5

PREDECESSORS
182 PLANT CONCEPT ENG PK

SUCCESSORS
2233 EST VAR EFFECTS REQ

TASK LABEL	TASK DESCRIPTION	DURATION (MONTHS)	START TIME	STOP TIME	SUCCESSORS
193	193 60% PROCESS PLANT PK	6.0	5.5	11.5	END EXPERIENCE
	PREDECESSORS				
	192 PLANT CONCEPT ENG PK				
	SUCCESSORS				
	214B 60% DESIGN QUAL ASM 214C 60% DESIGN QUANT AS 216B 60% DESIGN ASSESSME				
222A	222A EST VAR EFFECTS REQ	1.0	6.0	7.0	
	PREDECESSORS				
	221E FINALIZE BASELINES				
	SUCCESSORS				
	222B EST VAR EFFECTS SYS				
223B	223B EST VAR EFFECTS REQ	2.0	6.5	8.5	
	PREDECESSORS				
	223A EST PLANT BASELINE				
	SUCCESSORS				
	223C EST VAR EFFECTS SYS				
222B	222B EST VAR EFFECTS SYS	2.0	7.0	9.0	
	PREDECESSORS				
	222A EST VAR EFFECTS REQ				
	SUCCESSORS				
	222C FINALIZE MODS DEFIN 224A SUMMARIZE COMPARISN				
236A	236A RECOMMENDATIONS/REP	3.0	7.0	10.0	
	PREDECESSORS				

E34A SURVEY RFQS

NETWORK DESCRIPTION

TEST DATA

TASK LABEL	TASK DESCRIPTION	DURATION (MONTHS)	START TIME	STOP TIME
214A	214A INITIAL RAM ASSESSM	4 0	8 0	12 0
	PREDECESSORS			
	213D CONTRACT-RAM MOD DU			
	192 PLANT CONCEPT ENG PK			
	SUCCESSORS			
	214B 60% DESIGN QUAL ASM			
	214C 60% DESIGN QUANT AS			
216A	216A INITIAL DYNAMICS AS	4 0	8 0	12 0
	PREDECESSORS			
	215D CONT-DYN MODEL DEVE			
	192 PLANT CONCEPT ENG PK			
	SUCCESSORS			
	216B 60% DESIGN ASSESSME			
214D	214D CONTRACT-RAM MOD AS	11 0	8 0	19 0
	PREDECESSORS			
	213D CONTRACT-RAM MOD DU			
	NO SUCCESSORS			
216C	216C CONT-DYN MODEL ASSE	14 0	8 0	22 0
	PREDECESSORS			
	215D CONT-DYN MODEL DEVE			
	NO SUCCESSORS			
223C	223C EST VAR EFFECTS SYS	2 0	8 5	10 5
	PREDECESSORS			
	223B EST VAR EFFECTS REQ			
	SUCCESSORS			
	223D FINALIZE PLANT MOD1			

224A SUMMARIZE COMPARISN			
222C FINALIZE MOD8 DEFIN	2 0	9 0	
PREDECESSORS			
222B EST VAR EFFECTS SYS			
SUCCESSORS			
224B PREPARE REPORT			

TASK LABEL	TASK DESCRIPTION	DURATION (MONTHS)	START TIME	STOP TIME
[illegible]	END EXPERIENCE	6.0	10.0	16.0
	PREDECESSORS			
	236A RECOMMENDATIONS/REP			
	SUCCESSORS			
	END TASKS			
223D	223D FINALIZE PLANT MODS	1.0	13.5	11.5
	PREDECESSORS			
	223C EST UAR EFFECTS SYS			
	SUCCESSORS			
	224B PREPARE REPORT			
224A	224A SUMMARIZE COMPARISN	1.0	16.5	11.5
	PREDECESSORS			
	222B EST UAR EFFECTS SYS 223C EST UAR EFFECTS SYS			
	SUCCESSORS			
	224B PREPARE REPORT			
224B	224B PREPARE REPORT	1.0	11.5	12.5
	PREDECESSORS			
	224A SUMMARIZE COMPARISN 222 [illegible]NALIZE MODS DEFIN 223D FINALIZE PLANT MODS			
	SUCCESSORS			
	END AP DESIGN			
214B	214B 60% DESIGN QUAL ASM	1.5	12.0	13.5
	PREDECESSORS			

214A INITIAL RAM ASSESSM
193 60% PROCESS PLANT PE

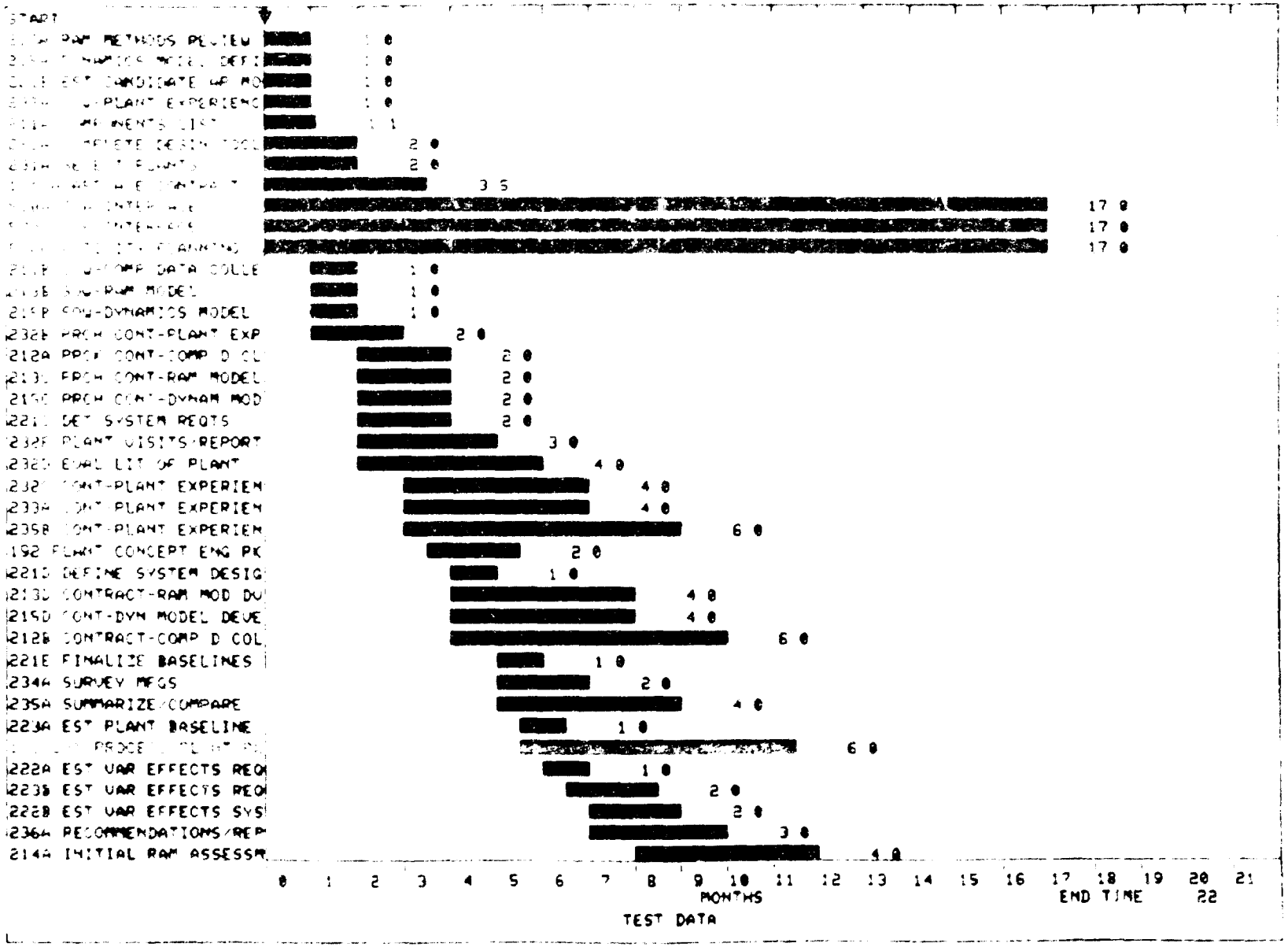
NO SUCCESSORS

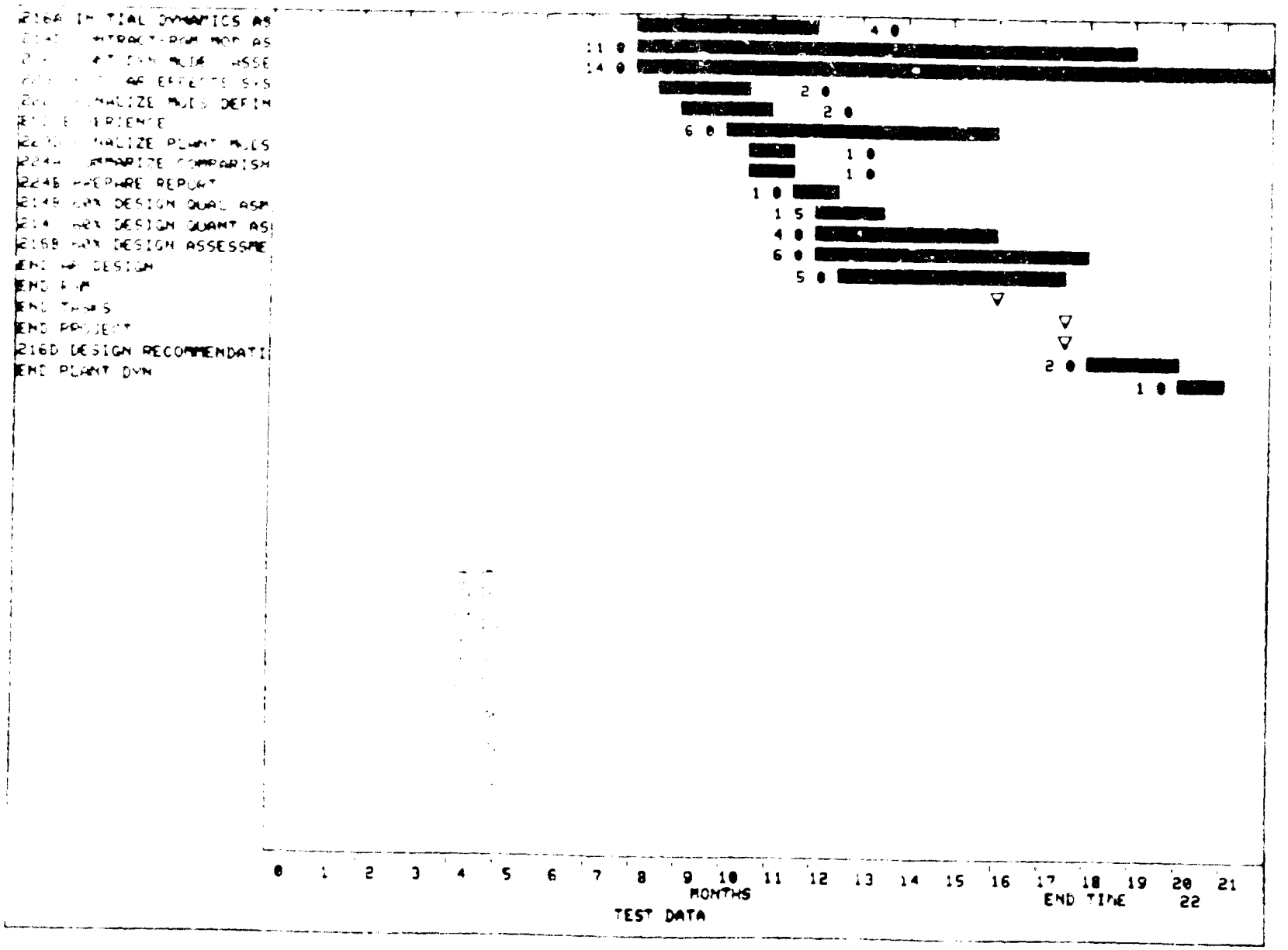
TASK LABEL	TASK DESCRIPTION	DURATION (MONTHS)	START TIME	STOP TIME	SUCCESSORS
214	214C 60% DESIGN QUANT AS	4.0	12.0	16.0	END AP DESIGN END EXPERIENCE
	PREDECESSORS 214A INITIAL RAM ASSESSM 193 60% PROCESS PLANT PR				
	SUCCESSORS END RAM				
216B	216B 60% DESIGN ASSESSME	6.0	12.0	18.0	
	PREDECESSORS 216A INITIAL DYNAMICS AS 193 60% PROCESS PLANT PR				
	SUCCESSORS 216D DESIGN RECOMMENDATI				
0003	END AP DESIGN	5.0	12.5	17.5	END PROJECT
	PREDECESSORS 224B PREPARE REPORT				
	SUCCESSORS END TASKS				
0001	END RAM	0	16.0	16.0	
	PREDECESSORS 2.4C 60% DESIGN QUANT AS				
	SUCCESSORS END TASKS				
0005	END TASKS	0	17.5	17.5	
	PREDECESSORS END RAM				

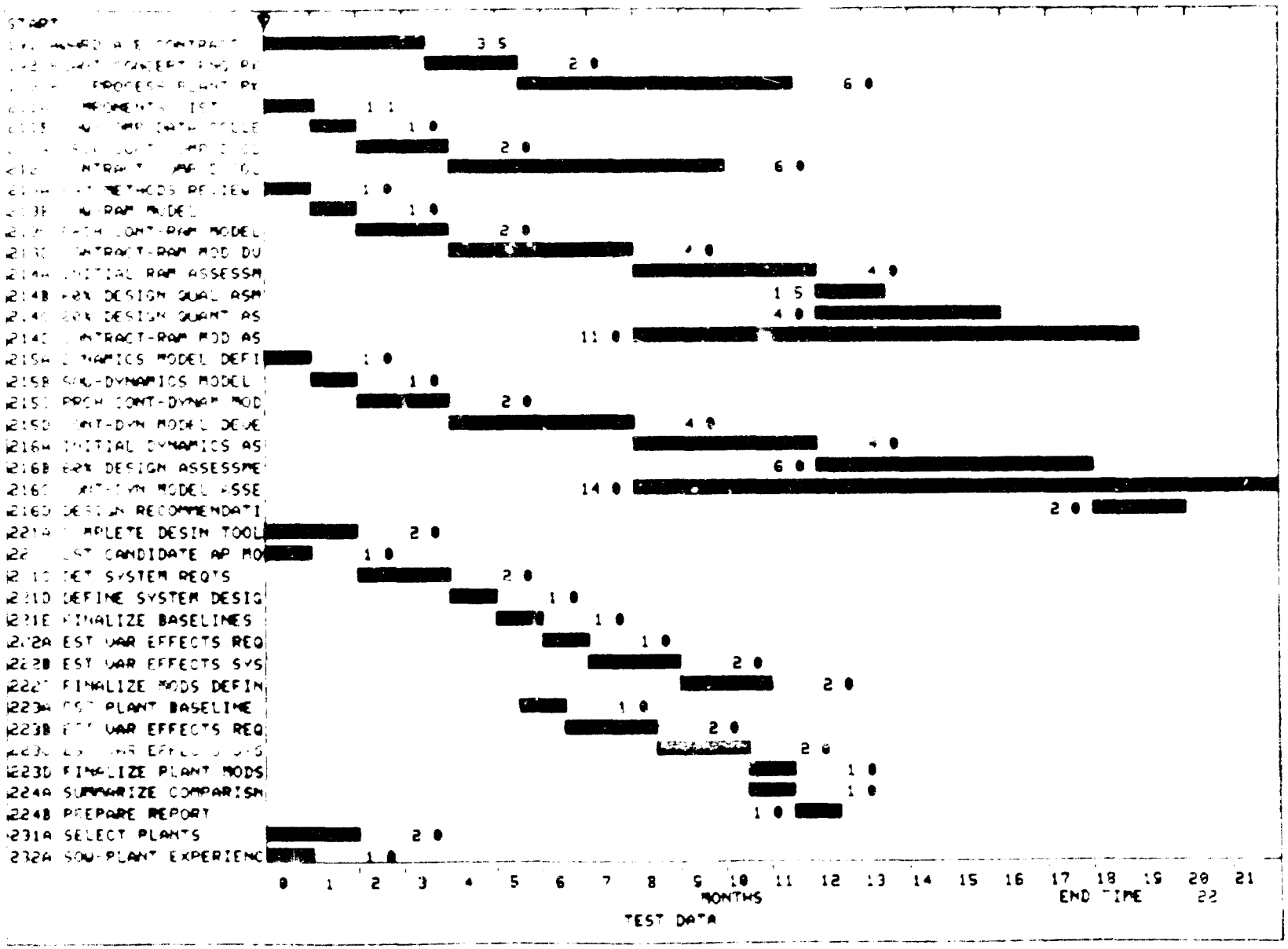
NETWORK DESCRIPTION

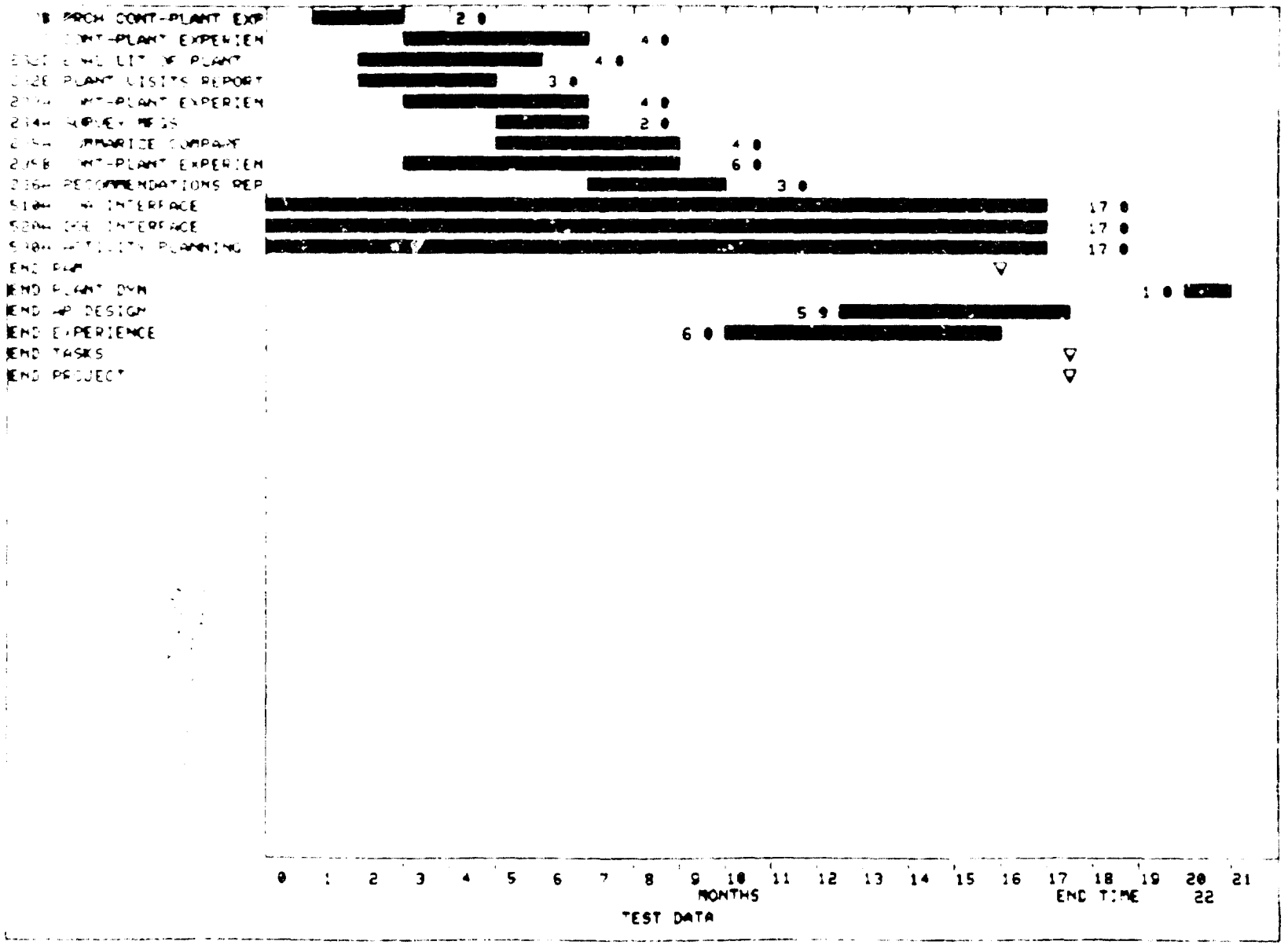
TEST DATA

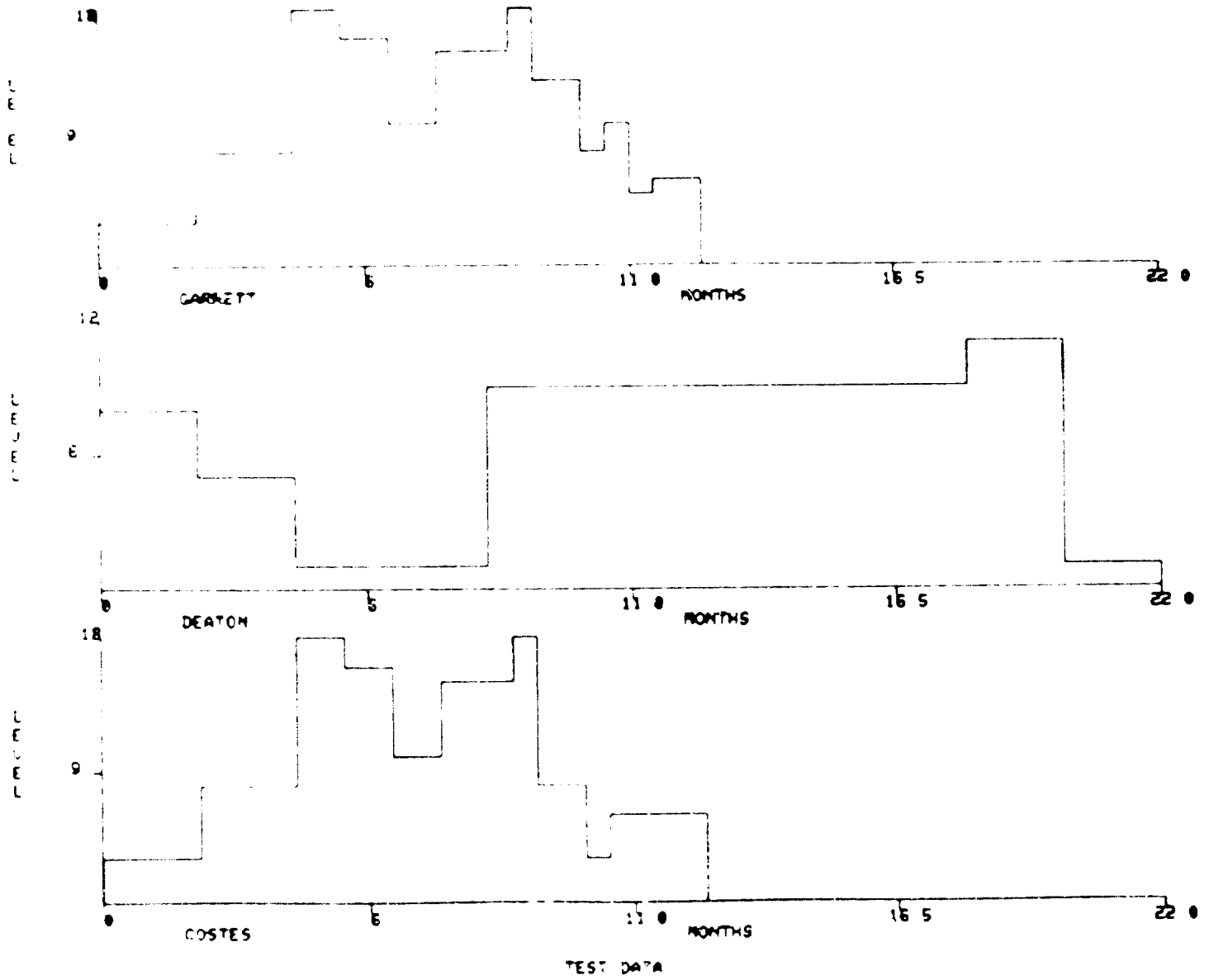
TASK CODE	TASK DESCRIPTION	DURATION (MONTHS)	START TIME	STOP TIME
0000	END PROJECT	0	17 S	17 S
	PREDECESSORS			
	END TASKS 5100 TOOL INTERFACE 5200 DOE INTERFACE 5300 ACTIVITY PLANNING			
	NO SUCCESSORS			
2160	2160 DESIGN RECOMMENDATI	2 0	18 0	20 0
	PREDECESSORS			
	2160 60% DESIGN ASSESSME			
	SUCCESSORS			
	END PLANT DYN			
0002	END PLANT DYN	1 0	20 0	21 0
	PREDECESSORS			
	2160 DESIGN RECOMMENDATI			
	NO SUCCESSORS			

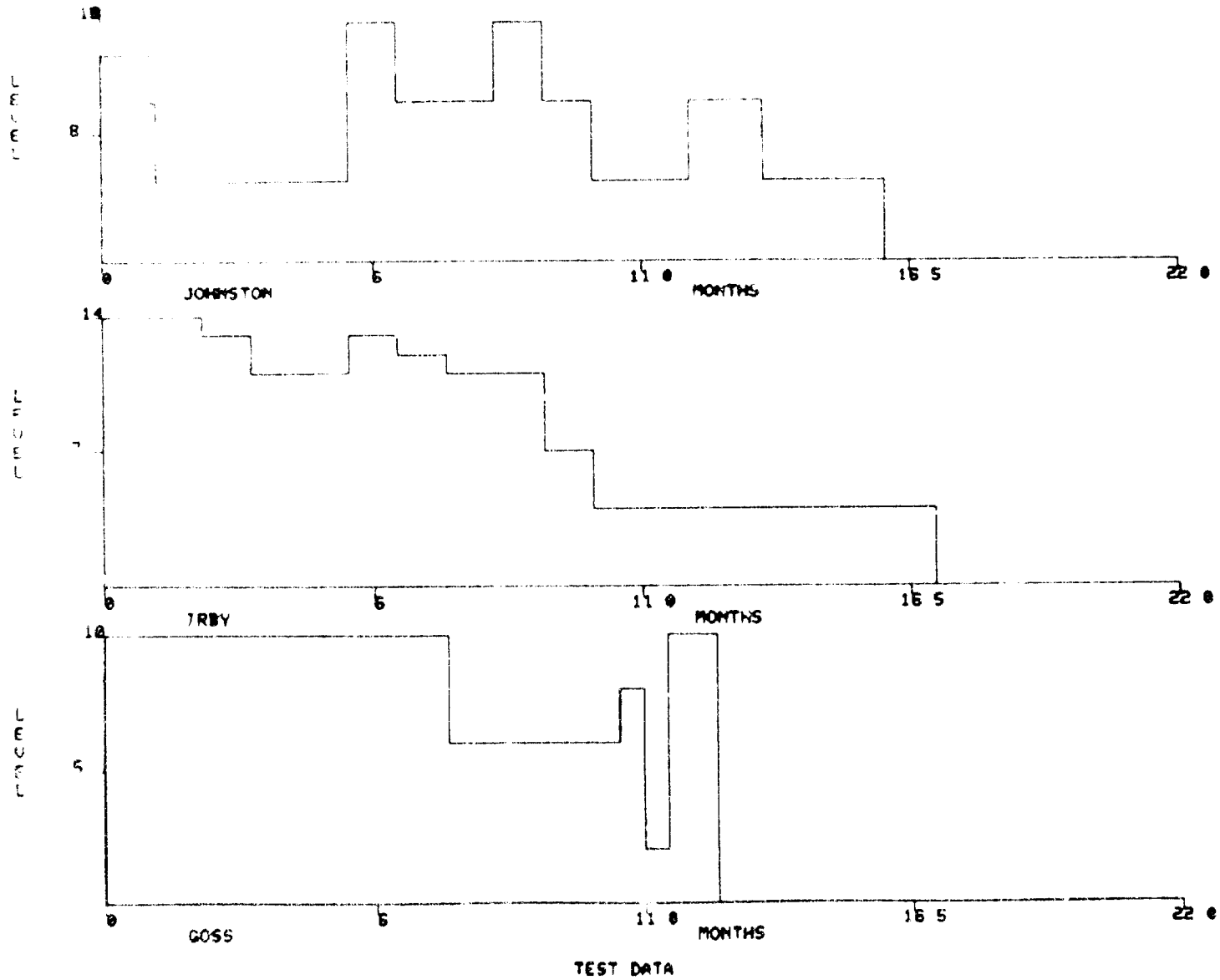

START
RAM METHODS REVIEW 1 0
DYNAMICS MODEL DEFI 1 0
EST CANDIDATE OF MO 1 0
PLANT EXPERIENC 1 0
MENTS LIST 1 1
DELETE DESIGN TOOL 2 0
PLANTS 2 0
3 5
17 0
17 0
17 0
COMP DATA COLLE 1 0
RAM MODEL 1 0
DYNAMICS MODEL 1 0
PROM CONT-PLANT EXP 2 0
PROM CONT-COMP D CL 2 0
PROM CONT-RAM MODEL 2 0
PROM CONT-DYNAM MOD 2 0
DET SYSTEM REQTS 2 0
PLANT VISITS/REPORT 3 0
EVAL LIT OF PLANT 4 0
CONT-PLANT EXPERIEN 4 0
CONT PLANT EXPERIEN 4 0
CONT-PLANT EXPERIEN 6 0
PLANT CONCEPT ENG PR 2 0
DEFINE SYSTEM DESIG 1 0
CONTRACT-RAM MOD DU 4 0
CONT-DYN MODEL DEVE 4 0
CONTRACT-COMP D COL 6 0
FINALIZE BASELINES 1 0
SURVEY MFGS 2 0
SUMMARIZE/COMPARE 4 0
EST PLANT BASELINE 1 0
6 0
EST VAR EFFECTS REQ 1 0
EST VAR EFFECTS REQ 2 0
EST VAR EFFECTS SYS 2 0
RECOMMENDATIONS/REP 3 0
INITIAL RAM ASSESSM 4 0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
MONTHS
END TIME 22
TEST DATA

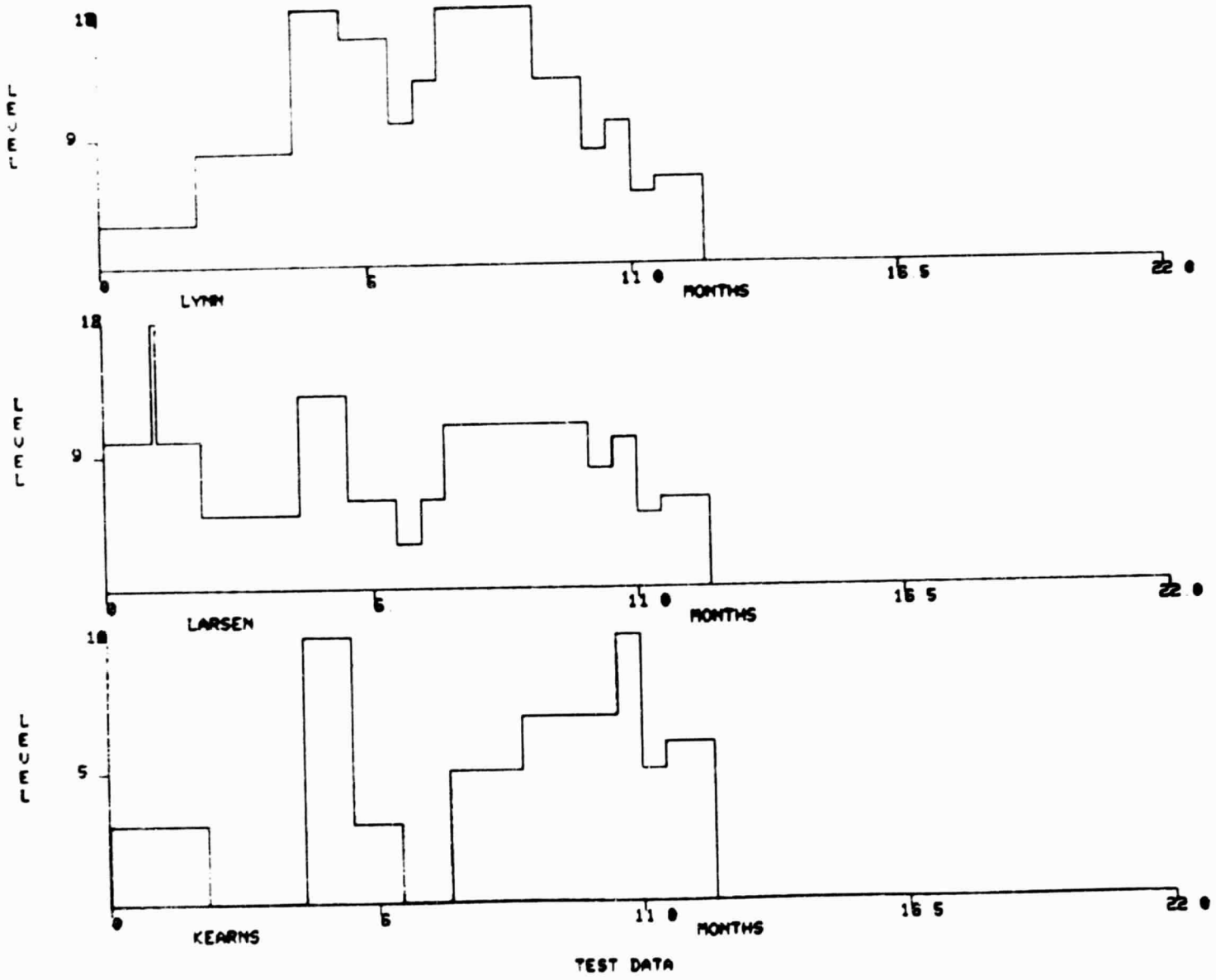

216A INITIAL DYNAMICS AS
216D DESIGN RECOMMENDATI
END PLANT DYN
4 0
11 0
14 0
2 0
2 0
6 0
1 0
1 0
1 0
1 5
4 0
6 0
5 0
2 0
1 0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
MONTHS
END TIME 22
TEST DATA

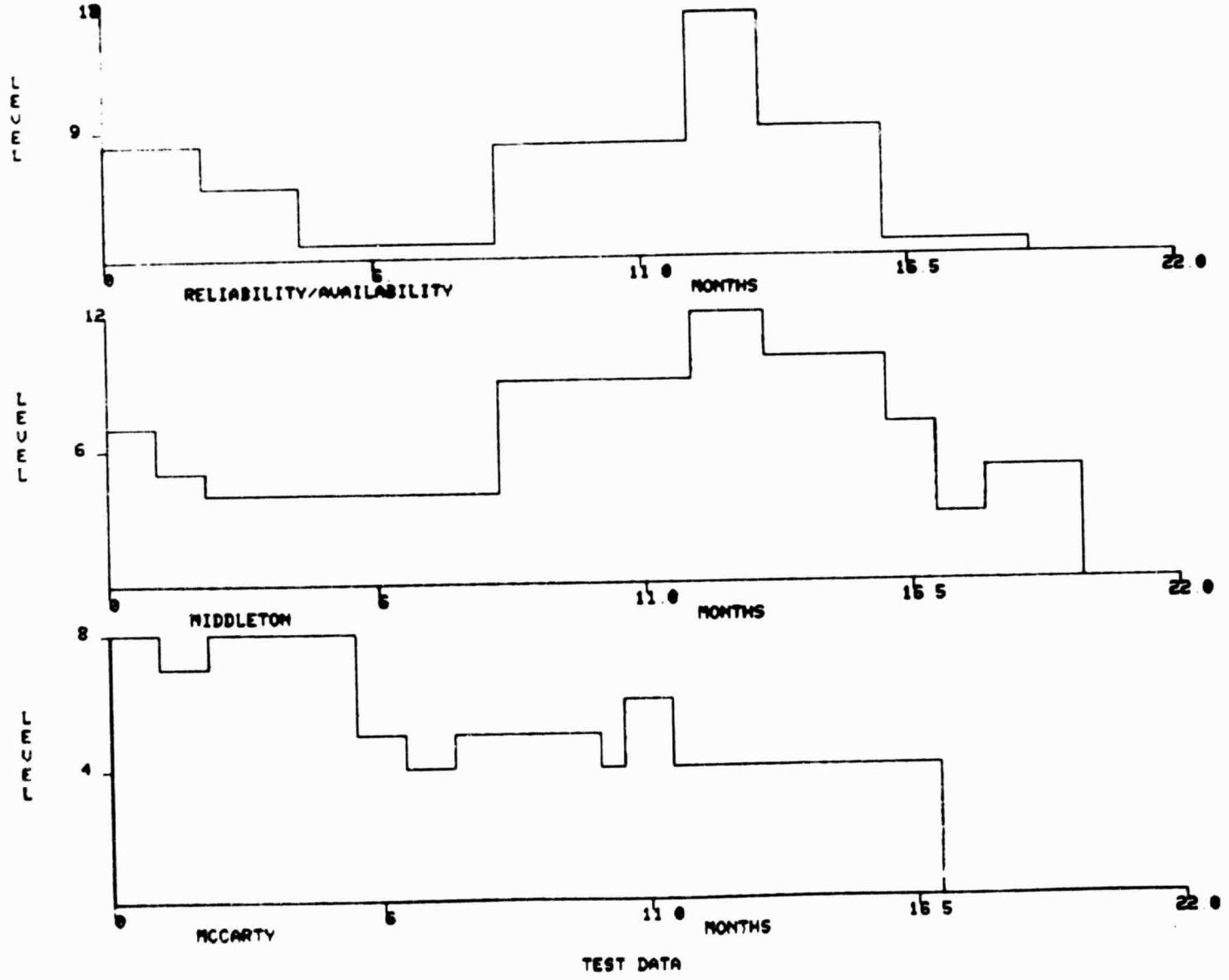

START
3 5
2 0
6 0
1 1
1 0
2 0
6 0
1 0
1 0
2 0
4 0
4 0
1 5
4 0
11 0
1 0
1 0
2 0
4 0
4 0
6 0
14 0
2 0
2 0
1 0
2 0
1 0
1 0
2 0
2 0
1 0
2 0
2 0
1 0
1 0
1 0
2 0
1 0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
MONTHS
END TIME 22
TEST DATA

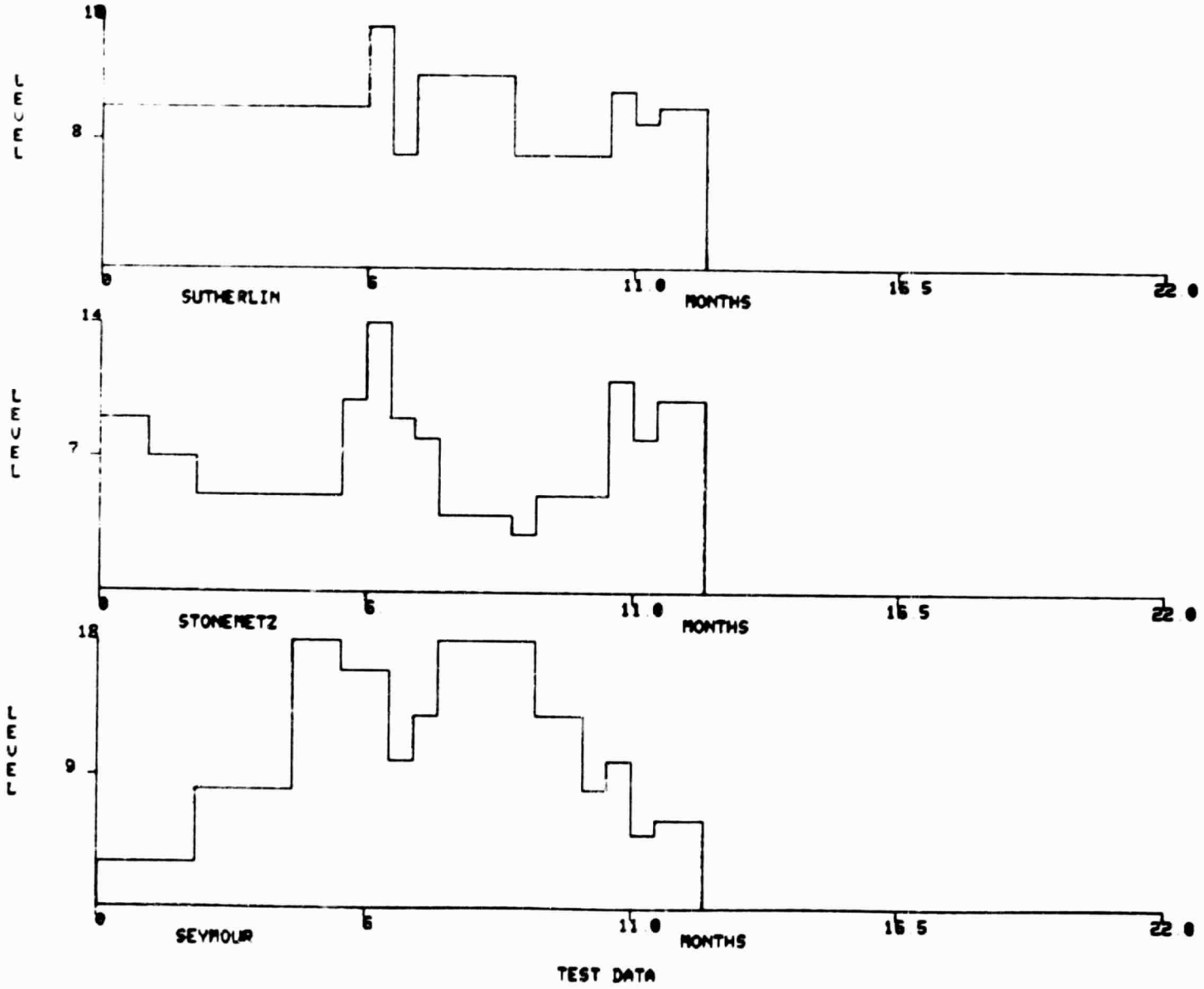

2 0
4 0
4 0
3 0
4 0
2 0
4 0
6 0
3 0
17 0
17 0
17 0
1 0
5 0
6 0
END PLANT DYN
END AP DESIGN
END EXPERIENCE
END TASKS
END PROJECT
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
MONTHS
END TIME 22
TEST DATA

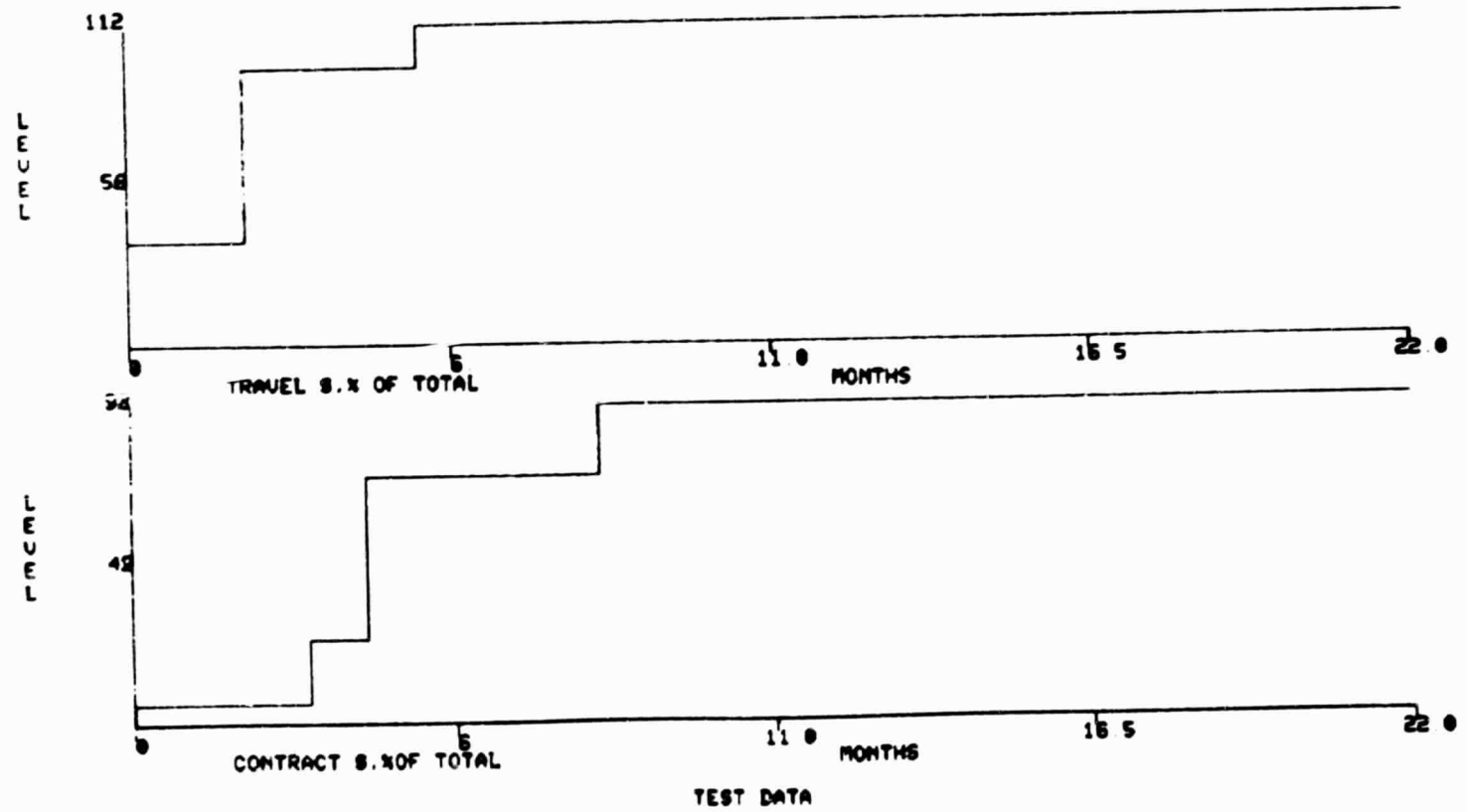

LEVEL
9
0
5
11 0
16 5
22 0
MONTHS
GARRETT
12
DEATON
COSTES
TEST DATA


8
0
5
11 0
16 5
22 0
MONTHS
JOHNSTON
14
7
TRBY
10
5
GOSS
TEST DATA


LEVEL
9
0
5
11 0
16 5
22 0
MONTHS
LYNN
LEVEL
9
0
5
11 0
16 5
22 0
MONTHS
LARSEN
LEVEL
5
0
5
11 0
16 5
22 0
MONTHS
KEARNS
TEST DATA


LEVEL
9
0
5
11.0
15.5
22.0
MONTHS
RELIABILITY/AVAILABILITY
LEVEL
12
6
0
5
11.0
15.5
22.0
MONTHS
MIDDLETON
LEVEL
8
4
0
5
11.0
15.5
22.0
MONTHS
MCCARTY
TEST DATA


LEVEL
8
SUTHERLIN
0
5
11 0
16 5
22 0
MONTHS
LEVEL
7
STONEMETZ
0
5
11 0
16 5
22 0
MONTHS
LEVEL
9
SEYMOUR
0
5
11 0
16 5
22 0
MONTHS
TEST DATA


112
56
LEVEL
0
5
11 0
16 5
22 0
TRAVEL $.% OF TOTAL
MONTHS
56
48
LEVEL
0
5
11 0
16 5
22 0
CONTRACT $.%OF TOTAL
MONTHS
TEST DATA

DATA FILE FOR CASE 2 - DATA2

```
  100 TABLE F-DEATON DATA
 1 000  GA01STOP OXYGEN FLOW                    0
 2 000  FP01START FIRE PREVENTION               0 0GA01
 3 000  GA02STOP COAL FLOW                      1 0GA01FP01
 4 000  UT01OPEN GASIFIER VENTS                 1 0GA01FP01
 5 030  CP01SURGE CONTROL COMPRESSOR           10 0UT01
 6 030  SL01STOP SLAG REMOVAL                  10 0GA02
 7 030  SG01STOP HP STEAM CIRCUIT               1 0UT01
 8 030  UT02CLOSE GASIFIER VENTS                1 0US01GA07
 9 000  SG02LP STEAM CIRC IN WHB               120 0GG01
10 030  AG01SURGE CONT ACID GAS REM            10 0UT01CP01
11 000  FP02STOP FIRE PREVENTION                1 0GG01FP01
12 000  GA03IGNITE PILOT BURNER                 0 0FP02SL01CH01
13 000  SG03BRING TEMP OF WHB DOWN             120 0GA03SL02
14 000  GA04TURN UP 7 GASIFIERS                 30 0AG01CP01
15 000  SL02STOP SLAG QUENCH                    10 0GA03
16 000  CH01STOP COAL FEED BIN FILLS            10 0GA02
17 000  SR01REGULATE SULFUR RECOVERY            10 0AG01
18 000  AG02TURN UP ACID GAS REMOVAL            35 0AG01CP01SR01
19 000  SR02TURN UP SULFUR RECOVERY             35 0SR01CP01
20 000  GA05STOP PILOT BURNER                   0 0SG02SG04
21 000  SG04STOP LP STEAM CIRC                  1 0SG03
22 000  GA06CONTINUE AIR FLOW                  30 0GA05
23 000  US01STOP WATER JACKET FLOW             30 0GA05
24 000  UT01REGULATE WASTE TREATMENT           30 0GA04
25 000  SD01REGULATE SOLIDS DISPOSAL           30 0GA04
26 000  GA07STOP AIR FLOW                       1 0GA06
27 000  GA08SHUTDOWN UNDER CONTROL              0 0UT02SD01UT01
28 000  0000STOP                                0 0AG02SR02GA06
29 000 EOF
```

```
'SET F:5 DP/DATA2
'START EXECUTE
```

```
                        MAIN MENU

SELECT ONE OF THE FOLLOWING

     1 - DEFINE AND LOAD NETWORK SOURCE FILE
     2 - EDIT NETWORK
     3 - PROCESS NETWORK
     4 - SET/RESET AUTO-HARDCOPY(CURRENTLY OFF )
     5 - SET/RESET SEGMENTATION OF PLOTS/TABLES(CURRENTLY OFF )
     6 - DISPLAY INPUT ERROR REPORT
     7 - DISPLAY NETWORK ECHO REPORT
     8 - DISPLAY RESOURCE ECHO REPORT
     9 - DISPLAY PREDECESSOR/SUCCESSOR REPORT
    10 - DISPLAY ACTIVITY SCHEDULING REPORT
    11 - DISPLAY RESOURCE HISTOGRAM TABLES
    12 - DISPLAY FULL TITLE TABLE
    13 - DISPLAY INPUT-ORDER TIMELINE PLOTS
    14 - DISPLAY WATERFALL ORDER TIMELINE PLOTS
    15 - DISPLAY RESOURCE HISTOGRAM PLOTS
    16 - DISPLAY ALL REPORTS AND PLOTS
    17 - DISPLAY ALL REPORTS
    18 - DISPLAY ALL PLOTS
     0 - TERMINATE

?01
```

```
 CHOOSE ONE OF THE FOLLOWING
   1 - DATA READ FROM SIGMA U BUILT FILE
   2 - DATA READ FROM IGDS BUILT FILE
?1
 WHAT IS THE TITLE OF THIS RUN-UP TO 36CHAR
?TEST DATA2
 UNITS TO BE USED
?MINUTE
```

```
                        MAIN MENU

SELECT ONE OF THE FOLLOWING

     1 - DEFINE AND LOAD NETWORK SOURCE FILE
     2 - EDIT NETWORK
     3 - PROCESS NETWORK
     4 - SET/RESET AUTO-HARDCOPY(CURRENTLY OFF )
     5 - SET/RESET SEGMENTATION OF PLOTS/TABLES(CURRENTLY OFF )
     6 - DISPLAY INPUT ERROR REPORT
     7 - DISPLAY NETWORK ECHO REPORT
     8 - DISPLAY RESOURCE ECHO REPORT
     9 - DISPLAY PREDECESSOR/SUCCESSOR REPORT
    10 - DISPLAY ACTIVITY SCHEDULING REPORT
    11 - DISPLAY RESOURCE HISTOGRAM TABLES
    12 - DISPLAY FULL TITLE TABLE
    13 - DISPLAY INPUT-ORDER TIMELINE PLOTS
    14 - DISPLAY WATERFALL ORDER TIMELINE PLOTS
    15 - DISPLAY RESOURCE HISTOGRAM PLOTS
    16 - DISPLAY ALL REPORTS AND PLOTS
    17 - DISPLAY ALL REPORTS
    18 - DISPLAY ALL PLOTS
     0 - TERMINATE

?03
```

```
                        MAIN MENU

SELECT ONE OF THE FOLLOWING

     1 - DEFINE AND LOAD NETWORK SOURCE FILE
     2 - EDIT NETWORK
     3 - PROCESS NETWORK
     4 - SET RESET AUTO-HARDCOPY(CURRENTLY OFF )
     5 - SET.RESET SEGMENTATION OF PLOTS/TABLES(CURRENTLY OFF )
     6 - DISPLAY INPUT ERROR REPORT
     7 - DISPLAY NETWORK ECHO REPORT
     8 - DISPLAY RESOURCE ECHO REPORT
     9 - DISPLAY PREDECESSOR/SUCCESSOR REPORT
    10 - DISPLAY ACTIVITY SCHEDULING REPORT
    11 - DISPLAY RESOURCE HISTOGRAM TABLES
    12 - DISPLAY FULL TITLE TABLE
    13 - DISPLAY INPUT-ORDER TIMELINE PLOTS
    14 - DISPLAY WATERFALL ORDER TIMELINE PLOTS
    15 - DISPLAY RESOURCE HISTOGRAM PLOTS
    16 - DISPLAY ALL REPORTS AND PLOTS
    17 - DISPLAY ALL REPORTS
    18 - DISPLAY ALL PLOTS
     0 - TERMINATE

?>7
```

TEST DATA8
SCHEDULING PROGRAM INPUT ECHO REPORT

SEQ NBR	TASK LABEL	TASK TITLE	DUR	NO PRED	PRED 1	PRED 2	PRED 3	PRED 4	PRED 5	NO S/R	S/R 1	S/R 2	S/R 3	S/R 4	S/R 5
1	GA01	STOP OXYGEN FLOW	0	0						0					
2	FP01	START FIRE PREVENTION	0	1	GA01					0					
3	GA02	STOP COAL FLOW	1	2	GA01	FP01				0					
4	VT01	OPEN GASIFIER VENTS	1	2	GA01	FP01				0					
5	CP01	SURGE CONTROL COMPRESSOR	10	1	VT01					0					
6	SL01	STOP SLAG REMOVAL	10	1	GA02					0					
7	SG01	STOP HP STEAM CIRCUIT	1	1	VT01					0					
8	VT02	CLOSE GASIFIER VENTS	1	2	WS01	GA07				0					
9	SG02	LP STEAM CIRC IN WHB	120	1	SG01					0					
10	AG01	SURGE CONT ACID GAS REM	10	2	VT01	CP01				0					
11	FP02	STOP FIRE PREVENTION	1	2	SG01	FP01				0					
12	GA03	IGNITE PILOT BURNER	0	3	FP02	SL01	CH01			0					
13	SG03	BRING TEMP OF WHB DOWN	120	2	GA03	SL02				0					
14	GA04	TURN UP 7 GASIFIERS	30	2	AG01	CP01				0					
15	SL02	STOP SLAG QUENCH	10	1	GA03					0					
16	CH01	STOP COAL FEED BIN FILLS	10	1	GA02					0					
17	SR01	REGULATE SULFUR RECOVERY	10	1	AG01					0					
18	AG02	TURN UP ACID GAS REMOVAL	35	3	AG01	CP01	SR01			0					
19	SR02	TURN UP SULFUR RECOVERY	35	2	SR01	CP01				0					
20	GA05	STOP PILOT BURNER	0	2	SG02	SG04				0					
21	SG04	STOP LP STEAM CIRC	1	1	SG03					0					
22	GA06	CONTINUE AIR FLOW	30	1	GA05					0					
23	WS01	STOP WATER JACKET FLOW	30	1	GA05					0					
24	UT01	REGULATE WASTE TREATMENT	30	1	GA04					0					
25	SD01	REGULATE SOLIDS DISPOSAL	30	1	GA04					0					
26	GA07	STOP AIR FLOW	1	1	GA06					0					
27	GA08	SHUTDOWN UNDER CONTROL	0	3	VT02	SD01	UT01			0					
28	0000	STOP	0	3	AG02	SR02	GA05			0					

MAIN MENU

SELECT ONE OF THE FOLLOWING

```
    1 - DEFINE AND LOAD NETWORK SOURCE FILE
    2 - EDIT NETWORK
    3 - PROCESS NETWORK
    4 - SET/RESET AUTO-HARDCOPY(CURRENTLY OFF )
    5 - SET/RESET SEGMENTATION OF PLOTS/TABLES(CURRENTLY OFF )
    6 - DISPLAY INPUT ERROR REPORT
    7 - DISPLAY NETWORK ECHO REPORT
    8 - DISPLAY RESOURCE ECHO REPORT
    9 - DISPLAY PREDECESSOR/SUCCESSOR REPORT
   10 - DISPLAY ACTIVITY SCHEDULING REPORT
   11 - DISPLAY RESOURCE HISTOGRAM TABLES
   12 - DISPLAY FULL TITLE TABLE
   13 - DISPLAY INPUT-ORDER TIMELINE PLOTS
   14 - DISPLAY WATERFALL ORDER TIMELINE PLOTS
   15 - DISPLAY RESOURCE HISTOGRAM PLOTS
   16 - DISPLAY ALL REPORTS AND PLOTS
   17 - DISPLAY ALL REPORTS
   18 - DISPLAY ALL PLOTS
    0 - TERMINATE
```

?09

TEST DATA2
PREDECESSOR-SUCCESSOR TABLE

SEQ NPP	TASK LABEL	TASK TITLE	NO PRED	PRED 1	PRED 2	PRED 3	PRED 4	PRED 5	PRED 6	PRED 7	NO SUCR	SUCR 1	SUCR 2	SUCR 3	SUCR 4	SUCR 5	SUCR 6	SUCR 7
1	CH01	STOP OXYGEN FLOW	0								3	FP01	GA02	UT01				
2	FP01	START FIRE PREVENTION	1	GA01							3	GA02	UT01	FP02				
3	CH02	STOP COAL FLOW	2	GA01	FP01						2	SL01	CH01					
4	UT01	OPEN GASIFIER VENTS	2	GA01	FP01						3	CP01	SG01	AG01				
5	CP01	SURGE CONTROL COMPRESSOR	1	UT01							4	AG01	GA04	AG02	SR02			
6	SL01	STOP SLAG REMOVAL	1	GA02							1	GA03						
7	SG01	STOP HP STEAM CIRCUIT	1	UT01							2	SG02	FP02					
8	UT02	CLOSE GASIFIER VENTS	2	US01	GA07						1	GA08						
9	SG02	LP STEAM CIRC IN WHB	1	SG01							1	GA05						
10	AG01	SURGE CONT ACID GAS REM	2	UT01	CP01						3	GA04	SR01	AG02				
11	FP02	STOP FIRE PREVENTION	2	SG01	FP01						1	GA03						
12	GA03	IGNITE PILOT BURNER	3	FP02	SL01	CH01					2	SG03	SL02					
13	SG03	BRING TEMP OF WHB DOWN	2	GA03	SL02						1	SG04						
14	GA04	TURN UP 7 GASIFIERS	2	AG01	CP01						2	UT01	SD01					
15	SL02	STOP SLAG QUENCH	1	GA03							1	SG03						
16	CH01	STOP COAL FEED BIN FILLS	1	GA02							1	GA03						
17	SR01	REGULATE SULFUR RECOVERY	1	AG01							2	AG02	SR02					
18	AG02	TURN UP ACID GAS REMOVAL	3	AG01	CP01	SR01					1	0000						
19	SR02	TURN UP SULFUR RECOVERY	2	SR01	CP01						1	0000						
20	GA05	STOP PILOT BURNER	2	SG02	SG04						2	GA06	US01					
21	SG04	STOP LP STEAM CIRC	1	SG03							1	GA05						
22	GA06	CONTINUE AIR FLOW	1	GA05							1	GA07						
23	US01	STOP WATER JACKET FLOW	1	GA05							1	UT02						
24	UT01	REGULATE WASTE TREATMENT	1	GA04							1	GA08						
25	SD01	REGULATE SOLIDS DISPOSAL	1	GA04							1	GA08						
26	GA07	STOP AIR FLOW	1	GA06							1	UT02						
27	GA08	SHUTDOWN UNDER CONTROL	3	UT02	SD01	UT01					1	0000						
28	0000	STOP	3	AG02	SR02	GA08					0							

?

```
                        MAIN MENU

SELECT ONE OF THE FOLLOWING

     1 - DEFINE AND LOAD NETWORK SOURCE FILE
     2 - EDIT NETWORK
     3 - PROCESS NETWORK
     4 - SET/RESET AUTO-HARDCOPY(CURRENTLY OFF )
     5 - SET/RESET SEGMENTATION OF PLOTS/TABLES(CURRENTLY OFF )
     6 - DISPLAY INPUT ERROR REPORT
     7 - DISPLAY NETWORK ECHO REPORT
     8 - DISPLAY RESOURCE ECHO REPORT
     9 - DISPLAY PREDECESSOR/SUCCESSOR REPORT
    10 - DISPLAY ACTIVITY SCHEDULING REPORT
    11 - DISPLAY RESOURCE HISTOGRAM TABLES
    12 - DISPLAY FULL TITLE TABLE
    13 - DISPLAY INPUT-ORDER TIMELINE PLOTS
    14 - DISPLAY WATERFALL ORDER TIMELINE PLOTS
    15 - DISPLAY RESOURCE HISTOGRAM PLOTS
    16 - DISPLAY ALL REPORTS AND PLOTS
    17 - DISPLAY ALL REPORTS
    18 - DISPLAY ALL PLOTS
     0 - TERMINATE

?10
```

TEST DATA2
TASK SCHEDULING TABLE

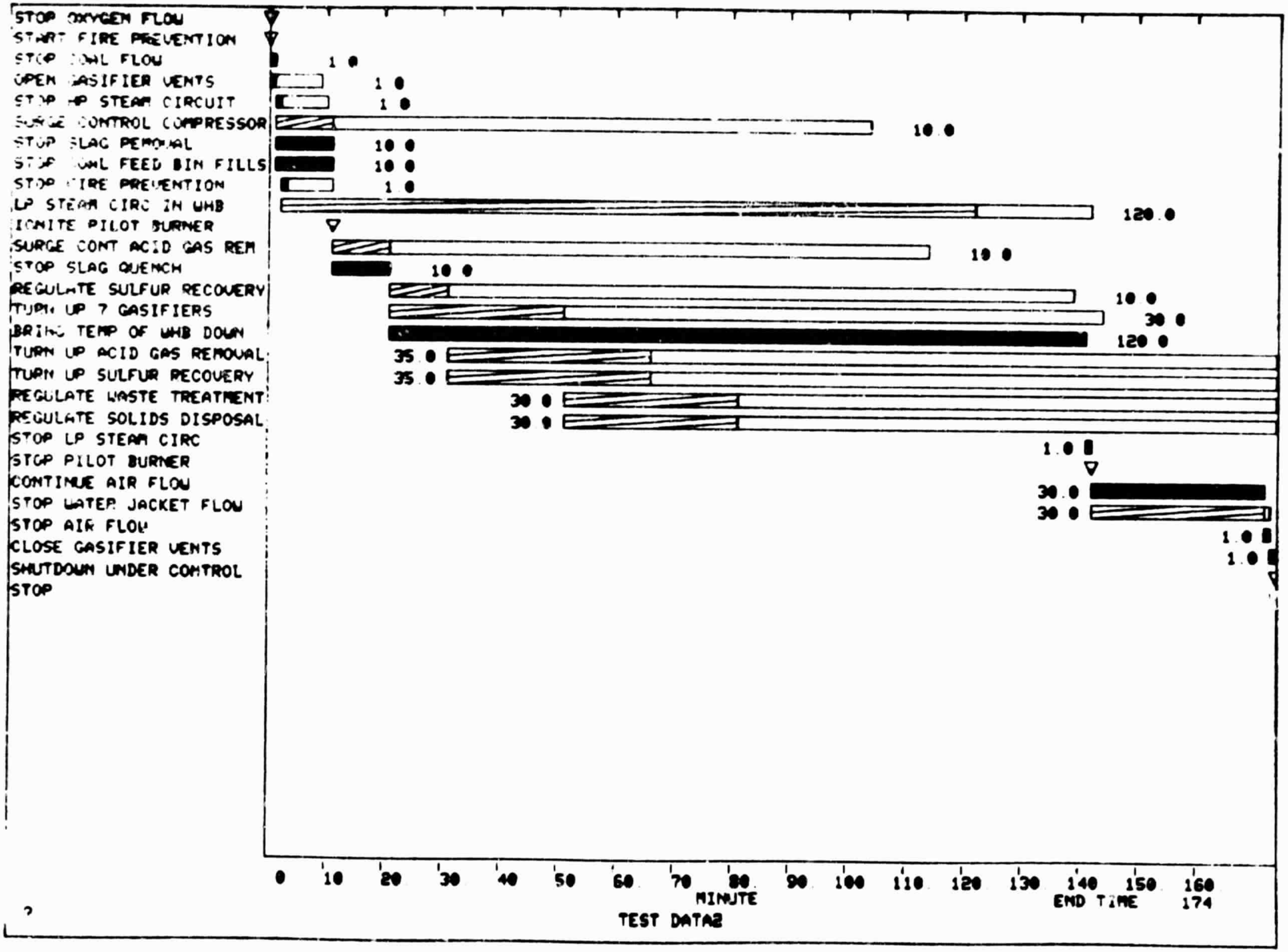
SEQ NBR	TASK LABEL	TASK TITLE	DUR	SHIFT	SCALED DUR	SLACK TIME	START TIME	STOP TIME
1	GA01	STOP OXYGEN FLOW	0	3	0	0	0	0
2	FP01	START FIRE PREVENTION	0	3	0	0	0	0
3	GA02	STOP COAL FLOW	1	3	1	0	0	1
4	VT01	OPEN GASIFIER VENTS	1	3	1	0	0	1
7	SG01	STOP HP STEAM CIRCUIT	1	3	1	0	1	2
5	CP01	SURGE CONTROL COMPRESSOR	10	3	10	0	1	11
6	SL01	STOP SLAG REMOVAL	10	3	10	0	1	11
16	CH01	STOP COAL FEED BIN FILLS	10	3	10	0	1	11
11	FP02	STOP FIRE PREVENTION	1	3	1	0	2	3
9	SG02	LP STEAM CIRC IN WHB	120	3	120	0	2	122
12	GA03	IGNITE PILOT BURNER	0	3	0	0	11	11
10	AG01	SURGE CONT ACID GAS REM	10	3	10	0	11	21
15	SL02	STOP SLAG QUENCH	10	3	10	0	11	21
17	SR01	REGULATE SULFUR RECOVERY	10	3	10	0	21	31
14	GA04	TURN UP 7 GASIFIERS	30	3	30	0	21	51
13	SG03	BRING TEMP OF WHB DOWN	120	3	120	0	21	141
18	AG02	TURN UP ACID GAS REMOVAL	35	3	35	0	31	66
19	SR02	TURN UP SULFUR RECOVERY	35	3	35	0	31	66
24	WT01	REGULATE WASTE TREATMENT	30	3	30	0	51	81
25	SD01	REGULATE SOLIDS DISPOSAL	30	3	30	0	51	81
21	SG04	STOP LP STEAM CIRC	1	3	1	0	141	142
20	GA05	STOP PILOT BURNER	0	3	0	0	142	142
22	GA06	CONTINUE AIR FLOW	30	3	30	0	142	172
23	WS01	STOP WATER JACKET FLOW	30	3	30	1	142	172
26	GA07	STOP AIR FLOW	1	3	1	0	172	173
8	VT02	CLOSE GASIFIER VENTS	1	3	1	0	173	174
27	GA08	SHUTDOWN UNDER CONTROL	0	3	0	0	174	174
28	0000	STOP	0	3	0	0	174	174

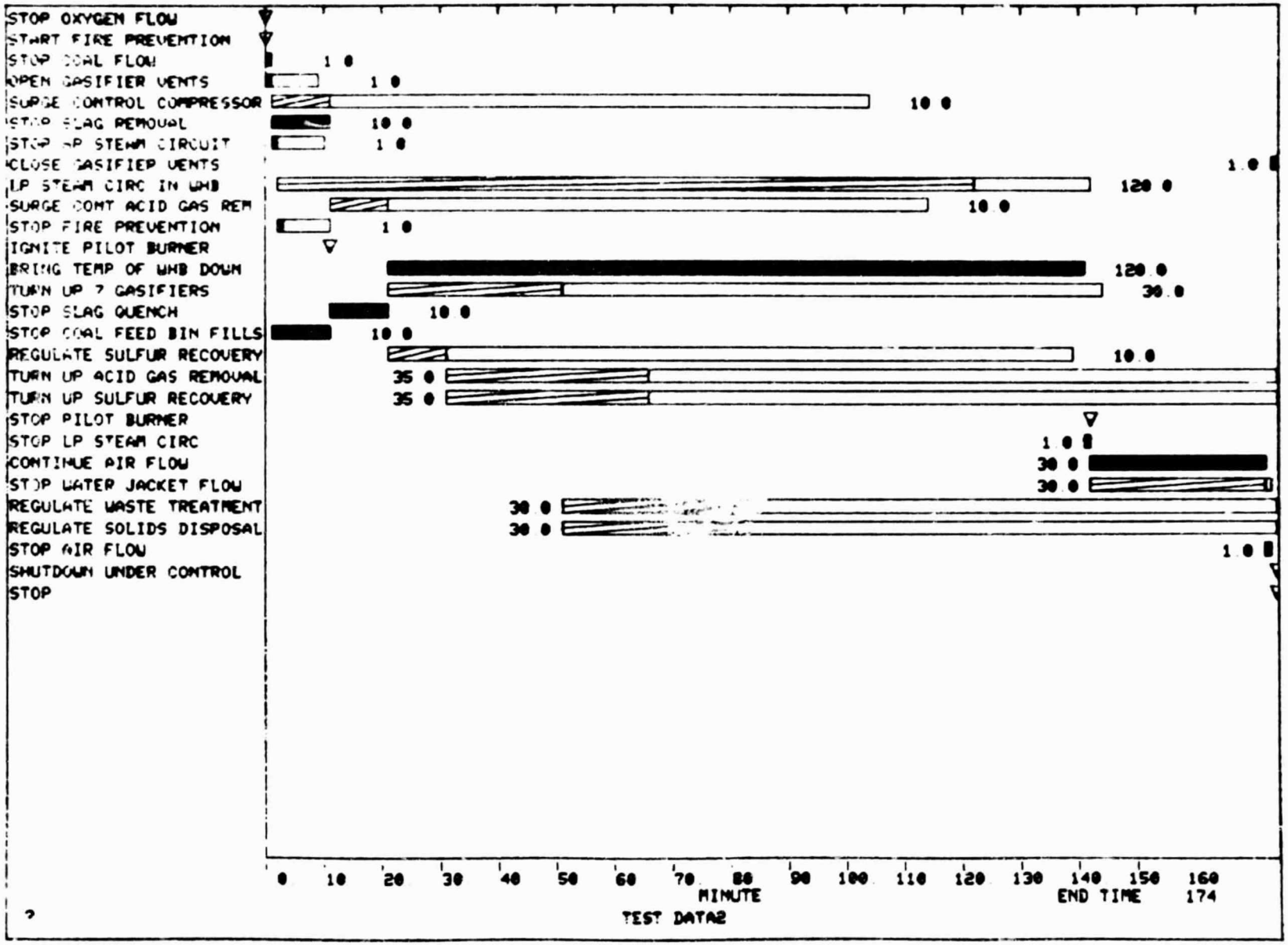
```
                              MAIN MENU

SELECT ONE OF THE FOLLOWING

     1 - DEFINE AND LOAD NETWORK SOURCE FILE
     2 - EDIT NETWORK
     3 - PROCESS NETWORK
     4 - SET/RESET AUTO-HARDCOPY(CURRENTLY OFF )
     5 - SET/RESET SEGMENTATION OF PLOTS/TABLES(CURRENTLY OFF )
     6 - DISPLAY INPUT ERROR REPORT
     7 - DISPLAY NETWORK ECHO REPORT
     8 - DISPLAY RESOURCE ECHO REPORT
     9 - DISPLAY PREDECESSOR/SUCCESSOR REPORT
    10 - DISPLAY ACTIVITY SCHEDULING REPORT
    11 - DISPLAY RESOURCE HISTOGRAM TABLES
    12 - DISPLAY FULL TITLE TABLE
    13 - DISPLAY INPUT-ORDER TIMELINE PLOTS
    14 - DISPLAY WATERFALL ORDER TIMELINE PLOTS
    15 - DISPLAY RESOURCE HISTOGRAM PLOTS
    16 - DISPLAY ALL REPORTS AND PLOTS
    17 - DISPLAY ALL REPORTS
    18 - DISPLAY ALL PLOTS
     0 - TERMINATE

?18
```




STOP OXYGEN FLOW
START FIRE PREVENTION
STOP COAL FLOW
OPEN GASIFIER VENTS
SURGE CONTROL COMPRESSOR
STOP SLAG REMOVAL
STOP LP STEAM CIRCUIT
CLOSE GASIFIER VENTS
LP STEAM CIRC IN WHB
SURGE CONT ACID GAS REM
STOP FIRE PREVENTION
IGNITE PILOT BURNER
BRING TEMP OF WHB DOWN
TURN UP 7 GASIFIERS
STOP SLAG QUENCH
STOP COAL FEED BIN FILLS
REGULATE SULFUR RECOVERY
TURN UP ACID GAS REMOVAL
TURN UP SULFUR RECOVERY
STOP PILOT BURNER
STOP LP STEAM CIRC
CONTINUE AIR FLOW
STOP WATER JACKET FLOW
REGULATE WASTE TREATMENT
REGULATE SOLIDS DISPOSAL
STOP AIR FLOW
SHUTDOWN UNDER CONTROL
STOP
1.0
10.0
120.0
30.0
35.0
30.0
0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160
MINUTE
END TIME 174
TEST DATA2

MAIN MENU

SELECT ONE OF THE FOLLOWING

1 - DEFINE AND LOAD NETWORK SOURCE FILE
2 - EDIT NETWORK
3 - PROCESS NETWORK
4 - SET/RESET AUTO-HARDCOPY(CURRENTLY OFF)
5 - SET/RESET SEGMENTATION OF PLOTS/TABLES(CURRENTLY OFF)
6 - DISPLAY INPUT ERROR REPORT
7 - DISPLAY NETWORK ECHO REPORT
8 - DISPLAY RESOURCE ECHO REPORT
9 - DISPLAY PREDECESSOR/SUCCESSOR REPORT
10 - DISPLAY ACTIVITY SCHEDULING REPORT
11 - DISPLAY RESOURCE HISTOGRAM TABLES
12 - DISPLAY FULL TITLE TABLE
13 - DISPLAY INPUT-ORDER TIMELINE PLOTS
14 - DISPLAY WATERFALL ORDER TIMELINE PLOTS
15 - DISPLAY RESOURCE HISTOGRAM PLOTS
16 - DISPLAY ALL REPORTS AND PLOTS
17 - DISPLAY ALL REPORTS
18 - DISPLAY ALL PLOTS
0 - TERMINATE

?15

```
THE PLOT AND TABLE SEGMENTATION IS CURRENTLY OFF
INPUT A 1 IF YOU WISH IT ON OR A 2 IF  YOU WISH IT OFF
?1
FOR INPUT-ORDER REPORTS AND PLOTS.
INPUT THE FIRST TASK IN THE SEQUENCE TO BE DISPLAYED
?A622
INPUT THE LAST TASK IN THE SEQUENCE TO BE DISPLAYED
?S001
FOR WATERFALL ORDER REPORTS AND PLOTS.
INPUT THE EARLIEST START TIME OF INTEREST
?20.
INPUT THE LATEST START TIME OF INTEREST
?,40
```

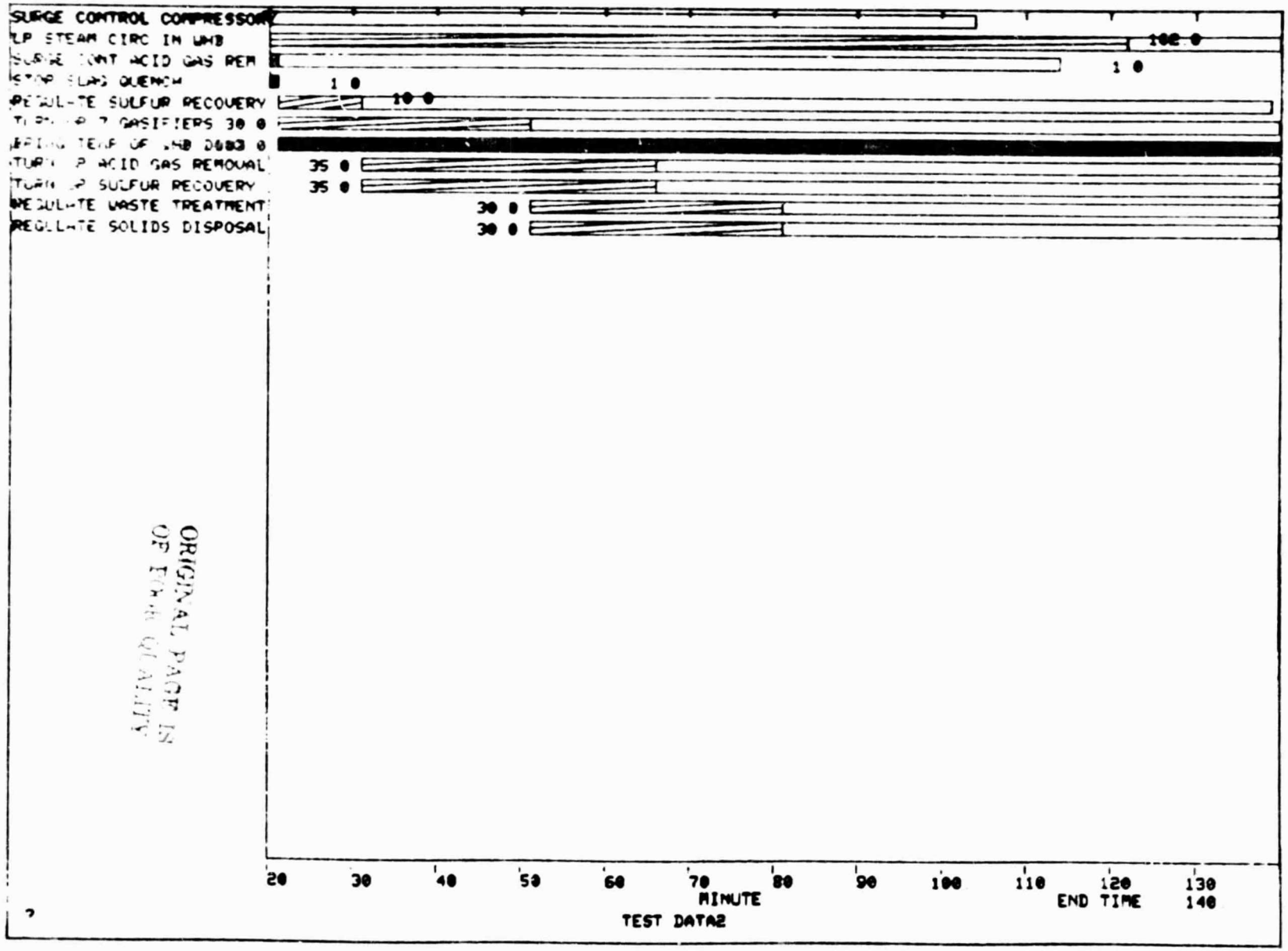
ORIGINAL PAGE IS
OF POOR QUALITY

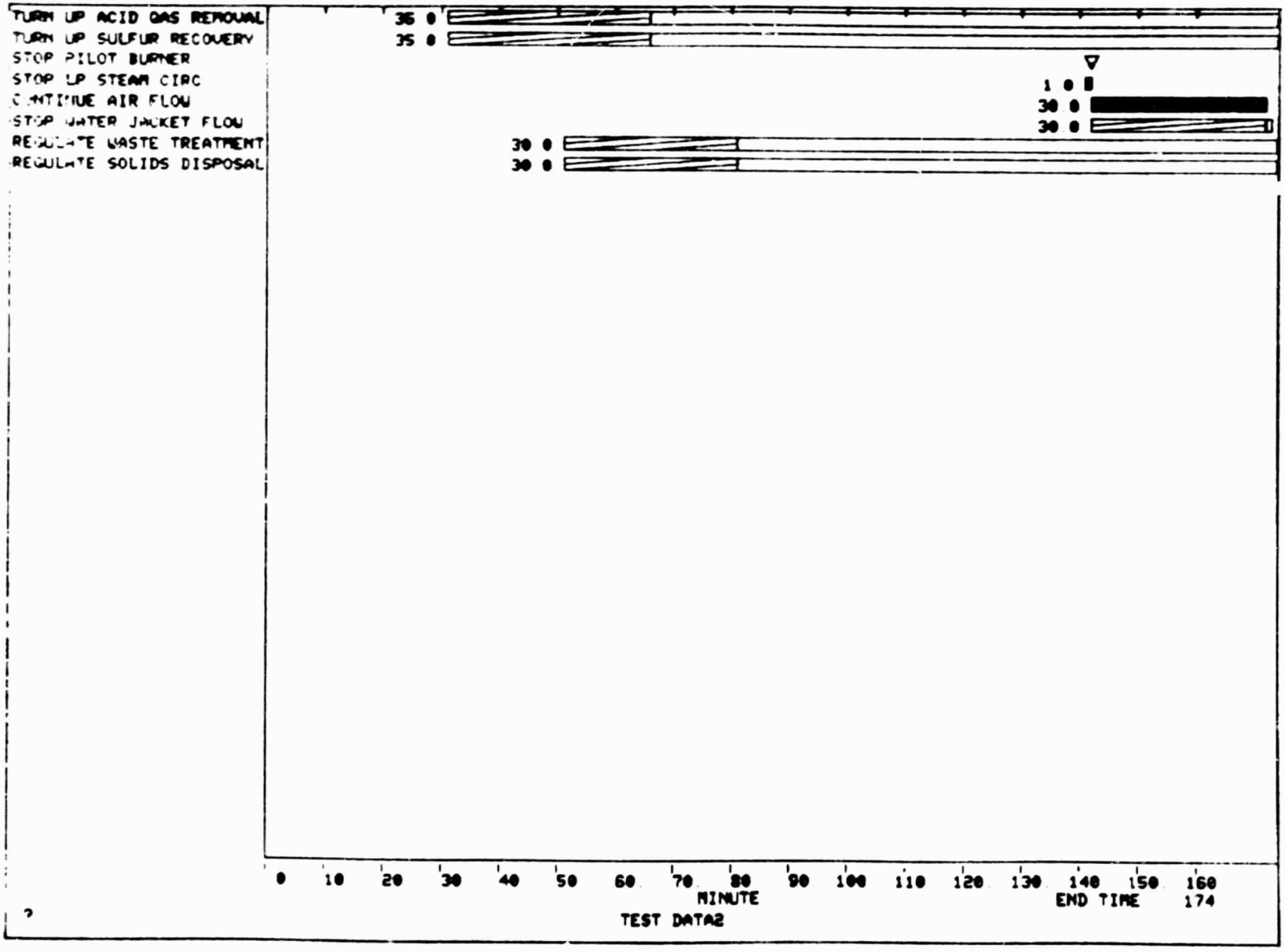
```
                              MAIN MENU

SELECT ONE OF THE FOLLOWING

     1 - DEFINE AND LOAD NETWORK SOURCE FILE
     2 - EDIT NETWORK
     3 - PROCESS NETWORK
     4 - SET/RESET AUTO-HARDCOPY(CURRENTLY OFF )
     5 - SET/RESET SEGMENTATION OF PLOTS/TABLES(CURRENTLY ON  )
     6 - DISPLAY INPUT ERROR REPORT
     7 - DISPLAY NETWORK ECHO REPORT
     8 - DISPLAY RESOURCE ECHO REPORT
     9 - DISPLAY PREDECESSOR/SUCCESSOR REPORT
    10 - DISPLAY ACTIVITY SCHEDULING REPORT
    11 - DISPLAY RESOURCE HISTOGRAM TABLES
    12 - DISPLAY FULL TITLE TABLE
    13 - DISPLAY INPUT-ORDER TIMELINE PLOTS
    14 - DISPLAY WATERFALL ORDER TIMELINE PLOTS
    15 - DISPLAY RESOURCE HISTOGRAM PLOTS
    16 - DISPLAY ALL REPORTS AND PLOTS
    17 - DISPLAY ALL REPORTS
    18 - DISPLAY ALL PLOTS
     0 - TERMINATE

?18
```




TURN UP ACID GAS REMOVAL
TURN UP SULFUR RECOVERY
STOP PILOT BURNER
STOP UP STEAM CIRC
CONTINUE AIR FLOW
STOP WATER JACKET FLOW
REGULATE WASTE TREATMENT
REGULATE SOLIDS DISPOSAL
35 0
35 0
1 0
30 0
30 0
30 0
30 0
0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160
MINUTE
END TIME 174
TEST DATA2

```
                        MAIN MENU

SELECT ONE OF THE FOLLOWING:

     1 - DEFINE AND LOAD NETWORK SOURCE FILE
     2 - EDIT NETWORK
     3 - PROCESS NETWORK
     4 - SET/RESET AUTO-HARDCOPY(CURRENTLY OFF )
     5 - SET/RESET SEGMENTATION OF PLOTS/TABLES(CURRENTLY ON  )
     6 - DISPLAY INPUT ERROR REPORT
     7 - DISPLAY NETWORK ECHO REPORT
     8 - DISPLAY RESOURCE ECHO REPORT
     9 - DISPLAY PREDECESSOR/SUCCESSOR REPORT
    10 - DISPLAY ACTIVITY SCHEDULING REPORT
    11 - DISPLAY RESOURCE HISTOGRAM TABLES
    12 - DISPLAY FULL TITLE TABLE
    13 - DISPLAY INPUT-ORDER TIMELINE PLOTS
    14 - DISPLAY WATERFALL ORDER TIMELINE PLOTS
    15 - DISPLAY RESOURCE HISTOGRAM PLOTS
    16 - DISPLAY ALL REPORTS AND PLOTS
    17 - DISPLAY ALL REPORTS
    18 - DISPLAY ALL PLOTS
     0 - TERMINATE

?0
```

APPENDIX A

SOURCE CODE LISTING

14:46 MAY 21, '81 DC/MATRIX.FAGUE

```
      1 -     1.000 C
      2 -     2.000           PARAMETER MAXLNK = 12
      3 -     3.000           PARAMETER MAXTSK = 300
      4 -     4.000           PARAMETER LHEAD = MAXTSK+1
      5 -     5.000           PARAMETER MAXRSC = 50
      6 -     6.000           PARAMETER MAXSR = 12
      7 -     7.000           PARAMETER MAXCOR = 2*MAXTSK
      8 -     8.000           PARAMETER LCHEAD = MAXCOR+1
      9 -     9.000 C
     10 -    10.000           COMMON /MATRIX/TASK(MAXTSK),DUR(MAXTSK),
     11 -    11.000      &            NPRED(MAXTSK),PRED(MAXTSK,MAXLNK),NSUCR(MAXTSK),
     12 -    13.000      &            SUCR(MAXTSK,MAXLNK),MARK(MAXTSK),STIME(MAXTSK),
     13 -    14.000      &            NC(LHEAD),SC(LHEAD),NTASKS,NMARKS,ISCHED(MAXTSK),
     14 -    16.000      &            EARLY,LATE,NCPTR,LINK1,LINK2,LPRED(MAXTSK,MAXLNK),
     15 -    17.000      &            LSUCR(MAXTSK,MAXLNK),IOPT(MAXTSK),LEARN(MAXTSK),
     16 -    19.000      &            LCYCLE(MAXTSK),ISHIFT(MAXTSK),
     17 -    20.000      &            NRSCS,NSP(MAXTSK),
     18 -    22.000      &            CTIME(MAXCOR),LCOR(MAXCOR),LC(LCHEAD),SHIFT(3),
     19 -    23.000      &            IXT,IFATAL,SDUR(MAXTSK),IRQUAN(MAXRSC),IRCOST(MAXRSC)
     20 -    25.000      &            ,ISRF,IPRTS,IECHO,MTABLE(10000),TES2
     21 -    26.000      &,           SLACK(MAXTSK),VINIT(MAXRSC),IFCN(MAXRSC),IDIAG,
     22 -    27.000      &            ISC(MAXTSK),HEAD,BLANK,SMCR(MAXRSC),VTYP(MAXRSC)
     23 -    28.000 C
     24 -    28.100           COMMON /MATRI/TSKLBL(MAXTSK,4),LSR(MAXTSK,MAXSR),RSC(MAXRSC),
     25 -    28.200      &            RSCLBL(MAXRSC,4),PTITLE(6),ITUNIT,ADUR(MAXTSK)
     26 -    28.300      &            ,IPFLG(MAXTSK)
     27 -    28.500           COMMON/SEG/WORDRF,WORDRL,IORDRF,IORDRL,ISEG,SEGMT(2)
     28 -    28.700           COMMON/RESCN/CNRS(2),CRESF(MAXRSC),CRESL(MAXRSC)
     29 -    29.000           INTEGER TASK,PRED,SUCR,NC,SC,HEAD,EARLY,NCPTR,SCPTR,BLANK
     30 -    29.500           INTEGER SMCR,VTYP,CRESF
     31 -    30.000           CHARACTER ADUR*5,LSR*8,TSKLBL*6
     32 -    31.000           CHARACTER  RSC*2,RSCLBL*6,ITUNIT*6,PTITLE*6
     33 -    31.500           CHARACTER ITES*1,ITESS*1,IET*1,IDTFS*1,IETES*1,IPFLG
     34 -    32.000 C
     35 -    33.000           EQUIVALENCE (NCPTR,SCPTR),(NULL,HEAD)
     36 -    33.500           EXTERNAL IHALFL,IHALFR,ISTRL,ISTRR
     37 -    34.000 C
     38 -    35.000           DATA NULL,EARLY,LATE,BLANK/LHEAD,1,2,'    '/,IFATAL/0/
     39 -    36.000           DATA LSR,RSCLBL/3600*'    ',200*'    '/
     40 -    37.000           DATA LPRED,LSUCR/3600*'    ',3600*'    '/
     41 -    37.500           DATA SHIFT/4.38,2.19,1.0/
     42 -    38.000           DATA NSP,SLACK/300*0,300*0/
     43 -    38.500                DATA NPRED,NSUCR/300*0,300*0/
     44 -    38.600                DATA PRED,SUCR/3600*'    ',3600*'    '/
     45 -    38.800           DATA SEGMT/'ON','OFF'/
     46 -    39.000           DATA IORDRF,IORDRL/'    ','    '/
     47 -    40.000           DATA CNRS/'ON','OFF'/
     48 -    41.000 C
```

```
1:47 MAY 21,'91 DC/TPLOTS.FAGUE

     1 -     1.000 C
     2 -     2.000 C
     3 -     3.000           COMMON /TPLOTS/ STRT(5),DURR(5),IFLG(5),DURX
     4 -     4.000      &           ,STIMM,STIM(2), ENDTM(2), TOT(2),XSIN, XS, JOB,
     5 -     5.000      &         NI, ILINE, STRTX, IFLGX,IPASS,IPASSX,DRXS1,DRXS2,
     6 -     6.000      &          XXS1,XSCALE(2),XTICSP(2)
     7 -     7.000 C
     8 -     7.500           COMMON /TPLOT/ NAME(4),ITITLE(6),IUNITS
     9 -     8.000           CHARACTER*6 NAME,ITITLE,IUNITS
    10 -     9.000 C
    11 -    10.000           DATA XSIN/11.5725/, ILINE/41/
```

14:47 MAY 21,'81 DC/ERRORS.FAGUE

```
1.000 C
2.000 C
3.000           COMMON/ERR/INTSK,IZ,IX,INRSC,ERTSK(300),ERRTSK(300)
3.500        *  ,ERLPD(300,11)
4.000 C
5.000           INTEGER ERTSK,ERRTSK,ERLPD
6.000 C
7.000           DATA ERLPD/3300*'    '/
8.000 C
```

C-2

```
14:47 MAY 21,'81 DC/COP.FAGUE

      1 -     1.000 C
      2 -     2.000 C         MATRIX CCP
      3 -     3.000           COMMON/CCP/ICPY
      4 -     4.000           DIMENSION COPYY(2)
      5 -     5.000           DATA COPYY/'ON','OFF'/
      6 -     6.000 C
```

```
:E MAY 21,'81 DC/TRAP6.FACOE

  1 -     1.000 C      MAIN ROUTINE-TRAP(SKILL)
  2 -     2.000        INCLUDE TPLOTS
  3 -     3.000        INCLUDE ERRORS
  4 -     4.000        INCLUDE MATRIX
  5 -     5.000        INCLUDE COP
  6 -     6.000 C
  7 -     7.000 C
  8 -     8.000        ICPY=2
  9 -     9.000        ISEG=2
 10 -    10.000        IPCESS=0
 11 -    11.000        IPUT=0
 12 -    12.000        CALL INIT
 13 -    13.000 C
 14 -    14.000   100 CONTINUE
 15 -    15.000        CALL CLEAR(3)
 16 -    16.000        WRITE(1,200)CCPYY(ICPY),SEGMT(ISEG)
 17 -    17.000   200 FORMAT(25X,'MAIN MENU'///1X,'SELECT ONE OF THE FOLLOWING:'//
 18 -    18.000       *5X,'1 - DEFINE AND LOAD NETWORK SOURCE FILE'/
 19 -    19.000       *5X,'2 - EDIT NETWORK'/
 20 -    20.000       *5X,'3 - PROCESS NETWORK'/
 21 -    21.000       *5X,'4 - SET/RESET AUTO-HARDCOPY(CURRENTLY',1X,A4,')'/
 22 -    22.000       *5X,'5 - SET/RESET SEGMENTATION OF PLOTS/TABLES(CURRENTLY',1X,
 23 -    23.000       *    A4,')'/
 24 -    24.000       *5X,'6 - DISPLAY INPUT ERROR REPORT'/
 25 -    25.000       *5X,'7 - DISPLAY NETWORK ECHO REPORT'/
 26 -    26.000       *5X,'8 - DISPLAY RESOURCE ECHO REPORT'/
 27 -    27.000       *5X,'9 - DISPLAY PREDECESSOR/SUCCESSOR REPORT'/
 28 -    28.000       *4X,'10 - DISPLAY ACTIVITY SCHEDULING REPORT'/
 29 -    29.000       *4X,'11 - DISPLAY RESOURCE HISTOGRAM TABLES'/
 30 -    30.000       *4X,'12 - DISPLAY FULL TITLE TABLE'/
 31 -    31.000       *4X,'13 - DISPLAY INPUT-ORDER TIMELINE PLOTS'/
 32 -    32.000       *4X,'14 - DISPLAY WATERFALL ORDER TIMELINE PLOTS'/
 33 -    33.000       *4X,'15 - DISPLAY RESOURCE HISTOGRAM PLOTS'/
 34 -    34.000       *4X,'16 - DISPLAY ALL REPORTS AND PLOTS'/
 35 -    35.000       *4X,'17 - DISPLAY ALL REPORTS'/
 36 -    36.000       *4X,'18 - DISPLAY ALL PLOTS'/
 37 -    37.000       *5X,'0 - TERMINATE'/)
 38 -    38.000 C
 39 -    39.000        READ(1,300)MM
 40 -    40.000   300 FORMAT(I2)
 41 -    41.000        CALL CLEAR(3)
 42 -    42.000        IF(MM.EQ.0) GO TO 400
 43 -    43.000        GO TO (1,2,3,4,17,5,6,6,8,8,8,8,11,11,13,14,15,16),MM
 44 -    44.000 C
 45 -    45.000     1 WRITE(1,250)
 46 -    46.000   250 FORMAT(1X,'CHOOSE ONE OF THE FOLLOWING:'/
 47 -    47.000       * 2X,'1 - DATA READ FROM SIGMA V BUILT FILE'/
 48 -    48.000       * 2X,'2 - DATA READ FROM IGDS BUILT FILE')
 49 -    49.000        READ(1,251)IPT
 50 -    50.000   251 FORMAT(I1)
```

```
 51 -    51.000        IF(IPT.EQ.1)CALL INPUT
 52 -    52.000        IF(IPT.EQ.2)CALL INPUT2
 53 -    53.000        IPUT=1
 54 -    54.000        GO TO 100
 55 -    55.000      2 CALL EDIT
 56 -    56.000        GO TO 100
 57 -    57.000      3 CALL PROCESS(IPUT)
 58 -    58.000        IPCESS=1
 59 -    59.000        GO TO 100
 60 -    60.000      4 CALL AUTOC
 61 -    61.000        GO TO 100
 62 -    62.000     17 CALL SEGMNT
 63 -    63.000        GO TO 100
 64 -    64.000      5 CALL INERR
 65 -    65.000        GO TO 100
 66 -    66.000      6 CALL ECHO(MM,IPCESS)
 67 -    67.000        GO TO 100
 68 -    68.000      8 CALL PRNTS(MM,IPCESS)
 69 -    69.000        GO TO 100
 70 -    70.000     11 CALL TIMLIN(MM,IPCESS)
 71 -    71.000        GO TO 100
 72 -    72.000     13 CALL HISTO(MM,IPCESS)
 73 -    73.000        GO TO 100
 74 -    74.000     14 CALL ECHO(MM,IPCESS)
 75 -    75.000        CALL PRNTS(MM,IPCESS)
 76 -    76.000        CALL TIMLIN(MM,IPCESS)
 77 -    77.000        CALL HISTO(MM,IPCESS)
 78 -    78.000        GO TO 100
 79 -    79.000     15 CALL ECHO(MM,IPCESS)
 80 -    80.000        CALL PRNTS(MM,IPCESS)
 81 -    81.000        GO TO 100
 82 -    82.000     16 CALL TIMLIN(MM,IPCESS)
 83 -    83.000        CALL HISTO(MM,IPCESS)
 84 -    84.000        GO TO 100
 85 -    85.000 C
 86 -    86.000 C      EXIT
 87 -    87.000    400 CONTINUE
 88 -    88.000        STOP
 89 -    89.000        END
 90 -    90.000        SUBROUTINE AUTOC
 91 -    91.000        INCLUDE COP
 92 -    92.000 C
 93 -    93.000 C      THIS ROUTINE SETS HDCOPY ON OR OFF
 94 -    94.000 C
 95 -    95.000        WRITE(1,10)COPYY(ICPY)
 96 -    96.000     10 FORMAT(' THE AUTOMATIC HARDCOPY IS CURRENTLY ',A3)
 97 -    97.000        WRITE(1,20)
 98 -    98.000     20 FORMAT(' INPUT A 1 IF YOU WISH IT ON OR A 2 IF YOU WISH IT OFF')
 99 -    99.000        READ(1,30)ICPY
100 -   100.000     30 FORMAT(I1)
101 -   101.000 C
102 -   102.000 C      EXIT
```

```
      RETURN
      END
      SUBROUTINE INERR
C
      INTEGER PRNTNW
      INCLUDE ERRORS
      INCLUDE MATRIX
C
C     THIS SUBROUTINE CHECKS AND PRINTS ERRORS FROM INPUT
C
      IF(INTSK.NE.0)PRINT 100
      IF(IZ.GT.1)PRINT 110
      IF(IX.GT.1)PRINT 120
      IF(IZ.EQ.1.AND.IZ.EQ.1)PRINT 130
      IF(ERRSC.NE.0)PRINT 150
      DO 10 I=1,NTASKS
           DO 20 J=1,MAXLNK
                IF(ERTSK(I).NE.0.AND.ERLPD(I,J).NE.BLANK)
     *          PRINT 160, ERTSK(I),ERLPD(I,J)
20         CONTINUE
      IF(ERRTSK(I).NE.0)PRINT 170, ERRTSK(I)
10    CONTINUE
C
      PRNTNW=1
      CALL RESRCE(PRNTNW)
C
100   FORMAT(' ***FATAL ERROR - NUMBER OF TASKS EXCEEDS AVAILABLE',
     *' STORAGE ***')
110   FORMAT(' ***FATAL ERROR - MORE THAN ONE CRITICAL JOB ',
     *'DESIGNATED')
120   FORMAT(' ***FATAL ERROR - MORE THAN ONE START TIME '
     *'DESIGNATED')
130   FORMAT(' ***FATAL ERROR - CRITICAL JOB AND START TIME',
     *' DESIGNATED')
150   FORMAT(' ***FATAL ERROR - NUMBER OF RESOURCES EXCEEDS',
     *' AVAILABLE STORAGE ***')
160   FORMAT(' ***FATAL ERROR - INVALID PREDECESSOR. ',
     *'TASK ',A4,', PRED. LABEL ',A4,' ***')
170   FORMAT(' ***FATAL ERROR - TASK ',A4,' HAS MORE '
     *'THAN THE MAXIMUM NUMBER OF SUCCESSORS ALLOWED ***')
C
      CALL WAIT(IQX,3,ICPY)
C
C
      RETURN
      END
      SUBROUTINE PROCESS(IPUT)
      INTEGER PRNTNW
      INCLUDE MATRIX
C
C     DETERMINE IF DATA HAS BEEN READ
      IF(IPUT.EQ.0)PRINT 10;CALL WAIT(I,3,))GO TO 999
```

```
155 -  155.000     10 FORMAT(' DATA HAS NOT BEEN READ, HIT RETURN AND'/
156 -  156.000       * ' CHOOSE A ''1'' IN THE MENU')
157 -  157.000 C
158 -  158.000         CALL SEQNCE
159 -  159.000         PRNTNW=0
160 -  160.000         CALL RESRCE(PRNTNW)
161 -  161.000         IF(IFATAL.EQ.1)PRINT 20;CALL WAIT(1,0,0);GO TO 999
162 -  162.000     20 FORMAT(' THERE ARE ERRORS IN THE NETWORK; HIT'/
163 -  163.000       *  ' RETURN AND CHOOSE A ''6'' IN THE MENU')
164 -  164.000 C
165 -  165.000 C      SCAN FORWARDS TILL ALL TASKS HAVE BEEN MARKED
166 -  166.000         CALL NTWKPS(EARLY)
167 -  167.000         CALL SCHED(EARLY)
168 -  168.000 C
169 -  169.000 C      SCAN BACKWARDS TILL ALL TASKS HAVE BEEN MARKED
170 -  170.000         CALL NTWKPS(LATE)
171 -  171.000         CALL SCHED(LATE)
172 -  172.000 C
173 -  173.000 C      BUILD STORED -CORNER LIST
174 -  174.000         CALL CORNER
175 -  175.000 C
176 -  176.000 C      BUILD RESOURCE HISTOGRAMS
177 -  177.000         CALL HIST
178 -  178.000 C
179 -  179.000 C      EXIT
180 -  180.000   999 CONTINUE
181 -  181.000         RETURN
182 -  182.000         END
183 -  183.000         SUBROUTINE EDIT
184 -  184.000 C
185 -  185.000 C
186 -  186.000         WRITE(1,10)
187 -  187.000     10 FORMAT(' THIS IS SUBROUTINE EDIT'/
188 -  188.000       &  ' HIT RETURN TO RETURN TO MENU & TERMINATE TO EDIT')
189 -  189.000         READ(1,20)I
190 -  190.000     20 FORMAT(I1)
191 -  191.000 C
192 -  192.000 C      EXIT
193 -  193.000         RETURN
194 -  194.000         END
195 -  195.000           SUBROUTINE ASK
196 -  196.000 C THIS SUBROUTINE ASKS ABOUT THE ECHO AND PRINT TABLES.
197 -  197.000           INCLUDE MATRIX
198 -  198.000           PRINT 10
199 -  199.000    10     FORMAT(' WHAT IS THE TITLE OF THIS RUN-UP TO 36CHAR')
200 -  200.000           READ (1,15)FTITLE
201 -  201.000    15     FORMAT(6A6)
202 -  202.000           PRINT 40
203 -  203.000    40     FORMAT(' UNITS TO BE USED')
204 -  204.000           READ(1,45)ITUNIT
205 -  205.000    45     FORMAT(A6)
206 -  206.000           RETURN
```

```
207 -  207.000          END
208 -  208.000          SUBROUTINE BAR(MARK,EARLY,LATE)
209 -  209.000 C
210 -  210.000          PARAMETER MAXTSK=300
211 -  211.000          INCLUDE TPLOTS
212 -  212.000          INCLUDE COP
213 -  213.000          COMMON/FLAG/IZB
214 -  214.000          INTEGER EARLY
215 -  215.000 C
216 -  216.000 C        SET UP LOOP TO PLOT TASK SYMBOL
217 -  217.000          IF(IZB.EQ.0)GOTO 10
218 -  218.000          ILINE = ILINE - 1
219 -  219.000          IF (ILINE.GE.1)GO TO 10
220 -  220.000 C
221 -  221.000 C        PAGE FILLED, WAIT, THEN DRAW NEW FRAME
222 -  222.000          CALL MOVEA(.2,.2)
223 -  223.000          CALL TSEND
224 -  224.000          CALL WAIT(ILINE,3,ICPY)
225 -  225.000          ILINE = 40
226 -  226.000          CALL FRAME
227 -  227.000 C
228 -  228.000     10   CONTINUE
229 -  229.000 C
230 -  230.000 C        LOOP TO PLOT ALL BARS ON LINE
231 -  231.000          IPASSX = 0
232 -  232.000          DO 50 I=1,IPASS
233 -  233.000          STRTX=STRT(I)
234 -  234.000          DURX=DURR(I)
235 -  235.000          IFLGX=IFLG(I)
236 -  236.000          IF(DURX .EQ. 0. ) GO TO 30
237 -  237.000 C
238 -  238.000 C        PLOT BAR
239 -  239.000     20   CONTINUE
240 -  240.000          IF(NAME(1) .EQ. ' ')GO TO 50
241 -  241.000          IPASSX = IPASSX + 1
242 -  242.000          CALL PLOTBR(MARK,EARLY,LATE)
243 -  243.000          GO TO 50
244 -  244.000 C
245 -  245.000 C        PLOT TRIANGLE
246 -  246.000     30   CONTINUE
247 -  247.000          IF(NAME(1) .EQ. ' ')GO TO 50
248 -  248.000          CALL TRNGLE
249 -  249.000     50   CONTINUE
250 -  250.000          IF(IFLG(3) .EQ. 4) CALL TRNGLE
251 -  251.000 C
252 -  252.000 C        EXIT
253 -  253.000          RETURN
254 -  254.000 C
255 -  255.000          END
256 -  256.000          SUBROUTINE CLEAR(ISIZE)
257 -  257.000 C
258 -  258.000 C  THIS SUBROUTINE CLEARS THE SCREEN AND RESETS THE CHARACTER SIZE
```

```
259 -  259.000 C
260 -  260.000 C          TEKTRONIX
261 -  261.000      10    CONTINUE
262 -  262.000            CALL BELL
263 -  263.000            CALL RESET
264 -  264.000            CALL CHRSIZ(ISIZE)
265 -  265.000            CALL NEWPAG
266 -  265.200            CALL IOWAIT(20)
267 -  265.400            DO 20, I=1,100000
268 -  265.500            IJ=I+10
269 -  265.600      20    CONTINUE
270 -  266.000            RETURN
271 -  267.000 C
272 -  268.000            END
273 -  269.000            SUBROUTINE COPY(LIST1,LIST2)
274 -  270.000 C
275 -  271.000            DIMENSION LIST1(1),LIST2(1)
276 -  272.000            INCLUDE MATRIX
277 -  273.000 C
278 -  274.000 C          MOVE TOP NODE OF LIST1 TO BOTTOM OF LIST2
279 -  275.000            NODE  = IHALFR(LIST1(HEAD))
280 -  276.000      10    CONTINUE
281 -  277.000            IF(NODE .EQ. NULL) GO TO 999
282 -  278.000            NEXT=IHALFR(LIST1(NODE))
283 -  279.000            CALL INSERT(NODE,HEAD, LIST2)
284 -  280.000            NODE  = NEXT
285 -  281.000            GO TO 10
286 -  282.000 C
287 -  283.000 C          EXIT
288 -  284.000    999     CONTINUE
289 -  285.000            RETURN
290 -  286.000 C
291 -  287.000            END
292 -  288.000            SUBROUTINE CORNER
293 -  289.000 C
294 -  290.000 C     THIS SUBROUTINE GENERATES A LIST OF ALL START AND STOP TIMES
295 -  291.000 C
296 -  292.000            INCLUDE MATRIX
297 -  293.000 C
298 -  294.000 C          INITIALIZE
299 -  295.000            LC(LCHEAD) = ISTRL(LCHEAD,LC(LCHEAD))
300 -  296.000            LC(LCHEAD) = ISTRR(LCHEAD,LC(LCHEAD))
301 -  297.000            IPTR = HEAD
302 -  298.000            ILAST = LCHEAD
303 -  299.000            INEXT = 1
304 -  300.000 C
305 -  301.000 C          SET UP STORED CORNER LIST
306 -  302.000            DO 10 I = 1,NTASKS
307 -  303.000 C
308 -  304.000 C          STORE TASK BEGIN INFORMATION IN NEXT NODE
309 -  305.000            IPTR = IHALFR(WC(IPTR))
310 -  306.000            STIME(IPTR)=STIME(IPTR)+TES2
```

```
311 -   307.000           CTIME(INEXT) = STIME(IPTR)
312 -   308.000           LCOR(INEXT) = ISTRL(0,LCOR(INEXT))
313 -   309.000           LCOR(INEXT) = ISTRR(IPTR,LCOR(INEXT))
314 -   310.000           CALL SORT(INEXT,ILAST,CTIME,CTIME,LC,LCHEAD)
315 -   311.000           INEXT = INEXT + 1
316 -   312.000 C
317 -   313.000 C         STORE TASK END INFORMATION IN NEXT NODE
318 -   314.000           CTIME(INEXT) = STIME(IPTR) + DUR(IPTR)
319 -   315.000           LCOR(INEXT) = ISTRL(1,LCOR(INEXT))
320 -   316.000           LCOR(INEXT) = ISTRR(IPTR,LCOR(INEXT))
321 -   317.000           CALL SORT(INEXT,ILAST,CTIME,CTIME,LC,LCHEAD)
322 -   318.000           INEXT = INEXT + 1
323 -   319.000    10     CONTINUE
324 -   320.000 C
325 -   321.000 C         EXIT
326 -   322.000   999     CONTINUE
327 -   323.000           RETURN
328 -   324.000 C
329 -   325.000           END
330 -   326.000           SUBROUTINE DEL(NODE,LIST)
331 -   327.000 C
332 -   328.000           DIMENSION LIST(1)
333 -   329.000 C
334 -   330.000 C         EXTRACT LINKS TO SUCCESSOR AND PREDICESSOR OF NODE
335 -   331.000           LKSUCR = IHALFR(LIST(NODE))
336 -   332.000           LKPRED = IHALFL(LIST(NODE))
337 -   333.000 C
338 -   334.000 C         RESET SUCCESSOR LINK IN PREDICESSOR
339 -   335.000           LIST(LKPRED) = ISTRR(LKSUCR,LIST(LKPRED))
340 -   336.000 C
341 -   337.000 C         RESET PREDICESSOR LINK IN SUCCESSOR NODE
342 -   338.000           LIST(LKSUCR) = ISTRL(LKPRED,LIST(LKSUCR))
343 -   339.000 C
344 -   340.000 C         ZERO LINKS IN DELETED NODE
345 -   341.000           LIST(NODE) = + 0
346 -   342.000 C
347 -   343.000 C         EXIT
348 -   344.000   999     CONTINUE
349 -   345.000           RETURN
350 -   346.000 C
351 -   347.000           END
352 -   348.000           SUBROUTINE ECHO(MM,IPCESS)
353 -   349.000 C
354 -   350.000 C     THIS SUBROUTINE PRINTS AN ECHO REPORT OF THE INPUT DATA
355 -   351.000 C
356 -   352.000           INCLUDE COF
357 -   353.000           INCLUDE MATRIX
358 -   354.000 C
359 -   355.000 C         DETERMINE IF DATA HAS BEEN PROCESSED
360 -   356.000           IF(IPCESS.EQ.0)PRINT 50;CALL WAIT(1,0,0);GO TO 999
361 -   357.000     50    FORMAT(' DATA HAS NOT BEEN PROCESSED, HIT RETURN'/
362 -   358.000       *     ' AND CHOOSE ''3'' IN THE MENU'/)
```

```
C
C       DETERMINE IF ERRORS IN INPUT
        IF(IFATAL.EQ.1)PRINT 51;CALL WAIT(I,0,0);GO TO 999
 51     FORMAT(' THERE ARE ERRORS IN INPUT; HIT RETURN AND'/
     *   ' CHOOSE A "6" IN THE MENU')
C
C       MAKE SURE IF SEGMENT FLAG IS ON INFORMATION IS GIVEN
        IF(ISEG.EQ.1.AND.IORDRF.EQ.BLANK)WRITE(1,52);
     *  CALL WAIT(I,0,0);GO TO 15
 52     FORMAT(' BEGINNING AND ENDING TASK NUMBERS WERE NOT GIVEN',
     *  'AND SEGMENT FLAG IS TURNED ON')
C
C       SKIP NETWORK ECHO LISTING IF ECHO FLAG IS SET TO 0
        IF(MM .EQ. 0) GO TO 15
C
C       PRINT HEADER INFORMATION
        CALL CLEAR(4)
        PRINT 100, (FTITLE(I),I=1,6)
100     FORMAT(/30X,6A6/30X,'SCHEDULING PROGRAM INPUT ECHO REPORT')
 41      PRINT 101
101     FORMAT(' SEQ  TASK',36X,'NO. ',5(' PRED'),5X,' NO.',
     &       5('   S/R')/' NBR  LABEL  TASK TITLE',
     &       T40,'DUR   PRED   1    2    3    4    5',6X,
     &       '  S/R    1     2     3     4     5'//)
        LINE = 0
C
C       PRINT DATA ENTRIES
        IQUTS=0
        DO 10 I = 1,NTASKS
        IF(ISEG.EQ.1)CALL CRTSKN(TASK(I),IQT,IQUTS,IORDRF,IORDRL)
        IF(IQT.EQ.1)GO TO 10
        PRINT 102, I,TASK(I),(TSKLBL(I,J),J=1,4),DUR(I),NPRED(I),
     &(LPRED(I,J),J=1,5),NSR(I),(LSR(I,J),J=1,5)
102     FORMAT(I4,2X,A4,3X,4A6,F5.0,I6,1X,5(1X,A4),I9,5(1X,A8))
        LINE = LINE + 1
        IF(LINE .GE. 50) CALL WAIT(LINE,4,ICRT)
 10     CONTINUE
        CALL WAIT(LINE,4,ICRT)
        IF(MM.NE.0.AND.MM.NE.16.AND.MM.NE.17)GO TO 999
C
C       PRINT RESOURCE TABLE
 15     CONTINUE
        IF(NRSCS .LT. 1) GO TO 999
        PRINT 104
104     FORMAT(//T5,'RESOURCE TABLE'//
     &       '  RSC   '/' CODE  DESCRIPTION'/)
        DO 20 I = 1,NRSCS
        PRINT 105, RSC(I),(RSCLBL(I,J),J=1,4)
105     FORMAT(2X,A2,3X,4A6)
        LINE = LINE + 1
        IF(LINE .GE. 50) CALL WAIT(LINE,4,ICRT)
 20     CONTINUE
```

```
C
C         WAIT FOR CUE BEFORE CONTINUING
   30     CONTINUE
          CALL WAIT(LINE,3,ICPY)
C
C         EXIT
  999     CONTINUE
          RETURN
C
          END
          SUBROUTINE HIST
C
C  THIS SUBROUTINE BUILDS THE HISTOGRAM TABLE FOR RESOURCES
C
          INCLUDE MATRIX
          CHARACTER IC*1,IR*2,IQ*5,ISIGN(4)*1,XIQ*5
          REAL MAXLVL
          DATA ISIGN/' ','*','+','-'/
C
C         SCAN CORNER LIST ONCE FOR EACH RESOURCE
          INDX = 1
          IC = ' '
          IQ = ' '
          XIQ = ' '
          IR = ' '
          DO 60 IRSC = 1,NRSCS
C
C         INITIALIZE HTABLE ENTRY
          HTABLE(INDX) = IRSC
          IN2 = INDX + 1
          INDX =  INDX + 4
          LINK1 = LCHEAD
          LK = IHALFR(LC(LINK1))
          TNOW = CTIME(LK)
          QLAST = 0.
          IQUAN = 0
          XIQUAN = 0.
          MAXLVL = 0
          NLVL = 0
C
C              SCAN ALL CORNER ENTRIES WITH SAME TIME
   10         CONTINUE
              LINK1 = IHALFR(LC(LINK1))
              IF(LINK1 .EQ. LCHEAD) GO TO 40
              TCOR = CTIME(LINK1)
              IF(TCOR .NE. TNOW) GO TO 40
C
C                  SCAN SEIZE RELEASE TABLE FOR THIS ENTRY
   20             CONTINUE
                  ISR = IHALFL(LCOR(LINK1))
                  ITSR = IHALFR(LCOR(LINK1))
                  INSR = NSR(ITSR)
```

```
467 -  463.000                    IF(INSR .EQ. 0) GO TO 10
468 -  464.000                    DO 30 I = 1,INSR
469 -  465.000                    IC=LSR(ITSK,I)(:1)
470 -  466.000                    IQ=LSR(ITSK,I)(2:6)
471 -  467.000                    IR=LSR(ITSK,I)(7:8)
472 -  468.000                    IF(IR .NE. RSC(IRSC)) GO TO 30
473 -  469.000                    DECODE(80,120,IQ) XNQ
474 -  470.000   120              FORMAT(F4.0)
475 -  471.000                    IF(VTYP(IRSC).EQ.1)GO TO 70
476 -  472.000                    NQ=XNQ
477 -  473.000                    IF(NQ .EQ. 0) NQ = 1
478 -  474.000                    IF(ISR .EQ. 1) GO TO 25
479 -  475.000                    IF(IFCN(IRSC).NE.0)IQUAN=RSCFCN(ITSK,IFCN,IRSC,TCOR);
480 -  476.000      *             GO TO 30
481 -  477.000                    IF(IC.EQ.ISIGN(3).OR.IC.EQ.ISIGN(1).OR.IC.EQ.ISIGN(2)
482 -  478.000      &           )IQUAN=IQUAN+NQ
483 -  479.000                   XIQUAN=IQUAN
484 -  480.000                          GO TO 30
485 -  481.000     25                   CONTINUE
486 -  482.000                          IF(IC.EQ.ISIGN(4).OR.IC.EQ.ISIGN(1).OR.IC.EQ.
487 -  483.000      *                    ISIGN(2))IQUAN=IQUAN-NQ
488 -  484.000                          XIQUAN=IQUAN
489 -  485.000                          GO TO 30
490 -  486.000     70           CONTINUE
491 -  487.000                  DECODE(80,121,IQ)XNQ
492 -  488.000     121          FORMAT(F5.1)
493 -  489.000                  IF(XNQ.EQ.0.)XNQ=1.
494 -  490.000                  IF(ISR.EQ.1)GO TO 26
495 -  491.000                  IF(IFCN(IRSC).NE.0)XIQUAN=RSCFCN(ITSK,IFCN,IRSC,TCOR);
496 -  492.000      *           GO TO 30
497 -  493.000                  IF(IC.EQ.ISIGN(3).OR.IC.EQ.ISIGN(1).OR.IC.EQ.ISIGN(2))
498 -  494.000      *           XIQUAN=XIQUAN+XNQ
499 -  495.000                  GO TO 30
500 -  496.000     26           CONTINUE
501 -  497.000                  IF(IC.EQ.ISIGN(4).OR.IC.EQ.ISIGN(1).OR.IC.EQ.ISIGN(2))
502 -  498.000      *           XIQUAN=XIQUAN-XNQ
503 -  499.000     30               CONTINUE
504 -  500.000                  GO TO 10
505 -  501.000 C
506 -  502.000 C                 TEST CURRENT LEVEL AGAINST LAST LEVEL
507 -  503.000     40           CONTINUE
508 -  504.000                  IF(XIQUAN .EQ. 0) GO TO 41
509 -  505.000                  HTABLE(INDX) = TNOW
510 -  506.000                  HTABLE(INDX+1) = XIQUAN + QLAST
511 -  507.000                  QLAST  = HTABLE(INDX+1)
512 -  508.000                  IF(QLAST  .GT. MAXLVL) MAXLVL = QLAST
513 -  509.000                   INDX = INDX + 2
514 -  510.000                  NLVL = NLVL - 1
515 -  511.000     41           CONTINUE
516 -  512.000                  IF(LINK1 .EQ. LCHEAD) GO TO 50
517 -  513.000                  IQUAN = 0
518 -  514.000                  XIQUAN = 0.
```

```
519 -  523.000                TROW = TCOR
520 -  524.000                GO TO 20
521 -  525.000 C
522 -  526.000 C          CLOSE HTABLE ENTRY
523 -  527.000     50     CONTINUE
524 -  528.000            HTABLE(IN2) = MAXLVL
525 -  529.000            HTABLE(IN2+1) = NLVL
526 -  530.000            HTABLE(IN2+2) = INDX
527 -  531.000     60      CONTINUE
528 -  532.000 C
529 -  533.000 C          RESOURCE HISTOGRAMS COMPLETE - PUT NULL MARKER IN POINTER
530 -  534.000            HTABLE(IN2+2) = 0
531 -  535.000 C
532 -  536.000 C          EXIT
533 -  537.000    999     CONTINUE
534 -  538.000            RETURN
535 -  539.000 C
536 -  540.000            END
537 -  541.000           SUBROUTINE HISTO(MM,IPCESS)
538 -  542.000 C
539 -  543.000            COMMON/HISTC/XFACT,YFACT,XINTV
540 -  544.000            COMMON/HIS/AME(5),IRTITL(4)
541 -  545.000            CHARACTER AME*1,IX*1,IRTITL*6
542 -  546.000          INTEGER ENDTIM,STIM,TOTIM
543 -  547.000          INCLUDE MATRIX
544 -  548.000          INCLUDE COP
545 -  549.000         DIMENSION NR(3),DATA(MAXTSK)
546 -  550.000         CHARACTER*6 IIUNIT
547 -  551.000          INCLUDE TPLOTS
548 -  552.000            DATA AME/'L','E','V','E','L'/
549 -  553.000 C
550 -  554.000 C          DETERMINE IF DATA HAS BEEN PROCESSED
551 -  555.000          IF(IPCESS.EQ.0)PRINT 31;CALL WAIT(1,0,0);GO TO 999
552 -  556.000     31   FORMAT(' DATA HAS NOT BEEN PROCESSED,HIT RETURN '/
553 -  557.000       *    ' AND CHOOSE ''3'' IN THE MENU')
554 -  558.000 C
555 -  559.000 C          DETERMINE IF ERRORS IN INPUT
556 -  560.000          IF(IFATAL.EQ.1)PRINT 32;CALL WAIT(1,0,0);GO TO 999
557 -  561.000     32   FORMAT(' THERE ARE ERRORS IN INPUT;HIT RETURN AND '/
558 -  562.000       *    ' CHOOSE A ''6'' IN THE MENU')
559 -  563.000 C
560 -  564.000 C          MAKE SURE IF SEGMENT FLAG IS TURNED ON INFORMATION IS GIVEN
561 -  565.000          IF(ISEG.EQ.1.AND.WORDRL.EQ.0.0)WRITE(1,33);
562 -  566.000       *  CALL WAIT(1,0,0);GO TO 999
563 -  567.000     33   FORMAT(' BEGINNING AND ENDING START TIMES WERE NOT GIVEN',
564 -  568.000       *    ' AND SEGMENT FLAG IS TURNED ON')
565 -  569.000 C
566 -  570.000 C          INITIALIZE THE HPLOT FACTORS
567 -  571.000     3      CONTINUE
568 -  572.000               DO 13 KG =1,6
569 -  573.000               ITITLE(KG) = FTITLE(KG)
570 -  574.000     13        CONTINUE
```

```
571 -  575.000            ITUNIT = ITUNIT
572 -  576.000 C
573 -  577.000            I = IHALFR(LC(LCHEAD))
574 -  578.000            STIMM    = CTIME(I)
575 -  579.000            I = IHALFL(LC(LCHEAD))
576 -  580.000            ENDTIM = CTIME(I)
577 -  581.000            IF(ISEG.NE.1)GO TO 34
578 -  582.000            STIMM=WORDRF
579 -  583.000            ENDTIM=WORDRL
580 -  584.000 C
581 -  585.000     34     TOTIM = ENDTIM - STIMM
582 -  586.000            XFACT = 11.47250/TOTIM
583 -  587.000            XINTV = TOTIM/4.
584 -  588.000 C
585 -  589.000 C          SELECT PLOT OPTION
586 -  590.000            IF(MM.EQ.16.OR.MM.EQ.18)GO TO 10
587 -  591.000      1     CONTINUE
588 -  592.000            PRINT 100
589 -  593.000    100     FORMAT(//' SELECT HISTOGRAM OPTION'/
590 -  594.000       &          5X,'1 - PLOT ALL RESOURCES'/
591 -  595.000       &          5X,'2 - PLOT SELECTED RESOURCES'/
592 -  596.000       &          5X,'0 - NO PLOTS')
593 -  597.000            READ(1,101) NPLOTS
594 -  598.000    101     FORMAT(I1)
595 -  599.000            IF(NPLOTS .EQ. 0) GO TO 999
596 -  600.000            IF(NPLOTS .EQ. 1) GO TO 10
597 -  601.000            IF(NPLOTS .EQ. 2) GO TO 20
598 -  602.000            GO TO 1
599 -  603.000 C
600 -  604.000 C          PLOT ALL HISTOGRAMS
601 -  605.000     10     CONTINUE
602 -  606.000            INDX = 1
603 -  607.000            LOCATN = 0
604 -  608.000               DO 15 IR = 1,NRSCS
605 -  609.000                  DO 11 J =1,4
606 -  610.000                  IRTITL(J) = RSCLBL(IR,J)
607 -  611.000       2         FORMAT(I1)
608 -  612.000      11           CONTINUE
609 -  613.000               IDATA = INDX + 4
610 -  614.000               MAXLVL = HTABLE(INDX+1)
611 -  615.000               NLVL = HTABLE(INDX+2)
612 -  616.000               INDX = HTABLE(INDX+3)
613 -  617.000               INMEN = MAXLVL + 1
614 -  618.000               INDMEN = INMEN/2
615 -  619.000               INMEN = INDMEN*2
616 -  620.000               CALL SCLY(INMEN)
617 -  621.000               INDMEN=INMEN/2
618 -  622.000               YFACT = 2.833/INMEN
619 -  623.000               NLV = 2*NLVL
620 -  624.000                  DO 888 IB=1,NLV
621 -  625.000                  MLVL =IDATA + IB - 1
622 -  626.000                  DATA(IB) = HTABLE(MLVL)
```

```
623 -  627.000   888          CONTINUE
624 -  628.000 C
625 -  629.000              CALL HPLOT(DATA,MLVL,INMEM,INDMEM,LOCATM,MRSCS,IIUNIT,ISEG,
626 -  630.000     *        WORDRF,VIKIT(IR))
627 -  631.000    15        CONTINUE
628 -  632.000              CALL WAIT(IXQ,3,ICPY)
629 -  633.000              GO TO 999
630 -  634.000 C
631 -  635.000 C         SELECT RESOURCES TO BE PLOTTED
632 -  636.000    20     CONTINUE
633 -  637.000           CALL CLEAR(2)
634 -  638.000           PRINT 120
635 -  639.000   120     FORMAT(' ENTER ''X'' TO LIST RESOURCES')
636 -  640.000           READ(1,121) IX
637 -  641.000   121     FORMAT(A1)
638 -  642.000           IF(IX .NE. 'X') GO TO 23
639 -  643.000              DO 22 I = 1,NRSCS
640 -  644.000              PRINT 122, I, RSC(I), (RSCLBL(I,J),J=1,4)
641 -  645.000   122        FORMAT(I5,5X,A2,5X,4A6)
642 -  646.000    22        CONTINUE
643 -  647.000    23 CONTINUE
644 -  648.000           PRINT 123
645 -  649.000   123     FORMAT(/' ENTER THE NUMBER OF RESOURCES TO BE PLOTTED ',/
646 -  650.000       &    ' MAX OF 3 AND ZERO FOR NO PLOTS')
647 -  651.000           READ(1,101) NPLOTS
648 -  652.000           IF (NPLOTS .EQ. 0) GO TO 999
649 -  653.000           PRINT 124
650 -  654.000   124     FORMAT(' ENTER INDICES FOR DESIRED RESOURCES')
651 -  655.000           READ(1,101) (NR(I),I=1,NPLOTS)
652 -  656.000 C
653 -  657.000 C         PLOT SELECTED HISTOGRAMS
654 -  658.000           LOCATM = 0
655 -  659.000              DO 30 I = 1,NPLOTS
656 -  660.000              K=NR(I)
657 -  661.000                 DO 12 J =1,4
658 -  662.000                 IRTITL(J) = RSCLBL(K,J)
659 -  663.000    12           CONTINUE
660 -  664.000              INDX = 1
661 -  665.000    25        CONTINUE
662 -  666.000              IR = HTABLE(INDX)
663 -  667.000              IF(IR .EQ. NR(I)) GO TO 26
664 -  668.000              INDX = HTABLE(INDX+3)
665 -  669.000              IF(INDX .NE. 0) GO TO 25
666 -  670.000              PRINT 125, NR(I)
667 -  671.000   125        FORMAT(I5,' IS NOT A VALID RESOURCE INDEX')
668 -  672.000              CALL WAIT(J,0,0)
669 -  673.000              GO TO 30
670 -  674.000    26        CONTINUE
671 -  675.000              IDATA = INDX + 1
672 -  676.000              MAXLVL = HTABLE(INDX+1)
673 -  677.000              NLVL = HTABLE(INDX+2)
674 -  678.000              INMEM = MAXLVL + 1
```

```
675 -  679.000              INDMEM = INMEM/2
676 -  680.000              INMEM = INDMEM*2
677 -  681.000              CALL SCLV(INMEM)
678 -  682.000              INDMEM=INMEM/2
679 -  683.000              YFACT = 2.833/INMEM
680 -  684.000              NVL =2*NLVL
681 -  685.000                  DO 777 IL=1,NVL
682 -  686.000                  MLV =IDATA + IL - 1
683 -  687.000                  DATA(IL) = HTABLE(MLV)
684 -  688.000    777           CONTINUE
685 -  689.000 C
686 -  690.000           CALL HPLOT(DATA,NLVL,INMEM,INDMEM,LOCATN,NPLOTS,IIUNIT,ISEG,
687 -  691.000       *   WORDRF,VINIT(IR))
688 -  692.000     30    CONTINUE
689 -  693.000           CALL WAIT(IXQ,3,ICPY)
690 -  694.000           GO TO 1
691 -  695.000 C
692 -  696.000 C      EXIT
693 -  697.000    999 CONTINUE
694 -  698.000       RETURN
695 -  699.000 C
696 -  700.000       END
697 -  701.000       SUBROUTINE HPLOT(TIMES,NLVL,INMEM,INDMEM,IPASSZ,ICOUNT,IIUNIT,ISEG
698 -  702.000      *,WORDRF,VINIT)
699 -  703.000 C
700 -  704.000       COMMON/HISTC/XFACT,YFACT,XINTV
701 -  705.000       COMMON/HIS/AME(5),IRTITL(4)
702 -  706.000       DIMENSION XSC(2)
703 -  707.000       DIMENSION TICY(5)
704 -  708.000       DIMENSION YSC(4),XTIC(15)
705 -  709.000       DIMENSION  Y(5),TIMES(4)
706 -  710.000       INTEGER DTITLX(9),DTILX(36),AMEN(1),AMENX(4),RTITLX(6),RTILX(24)
707 -  711.000       INTEGER IIUNIX(2),IUNIX(8),ANMEN(1),ANMENX(4),AMEX(1),AMEXX(4)
708 -  712.000       CHARACTER AME*1,IRTITL*6,PRTDAT*10,DATE*6,TIME*6,IIUNIT*6
709 -  713.000       CHARACTER*36 DTITLE
710 -  714.000       CHARACTER*24 RTITLE
711 -  715.000       INCLUDE TPLOTS
712 -  716.000       INCLUDE COP
713 -  717.000       DATA YSC/.85,4.183,7.606,2.833/
714 -  718.000          DATA PRTDAT/'      19  '/
715 -  719.000       DATA XSC/1.27576,12.74826/
716 -  720.000 C
717 -  721.000 C      INITIALIZE PLOT ROUTINES
718 -  722.000 C
719 -  723.000       IPASSZ = IPASSZ + 1
720 -  724.000       IF(IPASSZ .NE. 1) GO TO 702
721 -  725.000       CALL CLEAR(3)
722 -  726.000       CALL SWINDO(0,4096,0,3120)
723 -  727.000       CALL VWINDO(0.,14.4,0.,10.5)
724 -  728.000 C
725 -  729.000 C      PRINT DATE
726 -  730.000 C
```

```
      CALL SCLOCK(DATE,TIME,ESEC,E60SEC)
         PRTDAT(:2)=DATE(:2)
         PRTDAT(4:5)=DATE(3:4)
         PRTDAT(9:10)=DATE(5:6)
         PRTDAT(3:3)='/'
         PRTDAT(6:6)='/'
      CALL MOVEA(.2,.15)
      CALL HLABEL(10,PRTDAT)
C
C     PRINT TITLE OF JOB
C
         DTITLE(1:6)=ITITLE(1)
         DTITLE(7:12)=ITITLE(2)
         DTITLE(13:18)=ITITLE(3)
         DTITLE(19:24)=ITITLE(4)
         DTITLE(25:30)=ITITLE(5)
         DTITLE(31:36)=ITITLE(6)
         ENCODE(36,10,DTITLX)DTITLE
   10    FORMAT(A36)
         CALL KAM2AS(36,DTITLX,DTILX)
         CALL MOVEA(6.2456,.1)
         CALL HLABEL(36,DTILX)
C
C     DETERMINE FRAME POSITION AND DRAW FRAME
C
  700 CONTINUE
C
      YTIC2 = YSC(IPASSZ)+YSC(4)
      CALL MOVEA(XSC(1),YTIC2)
      CALL DRAWA(XSC(1),YSC(IPASSZ))
      CALL DRAWA(XSC(2),YSC(IPASSZ))
C
C     DRAW TIC MARKS AND LABEL PLOT
C
      XTICX = 0.1098
      YTIC = YSC(IPASSZ) - XTICX
      XTIC2 = XSC(1) - XTICX
      YTIC3 = YTIC - XTICX
      TICMRK = STIMM
C
      CALL MOVEA(XSC(1), YSC(IPASSZ))
      CALL DRAWA(XSC(1), YTIC)
      ENCODE(4,800,AMEN)  TICMRK
      CALL KAM2AS(3,AMEN,AMENX)
      CALL MOVEA(XTIC2,YTIC3)
      CALL HLABEL(3,AMENX)
C
      XTICVL = (XSC(2) - XSC(1))/4
      XTIC(3) = XSC(1) + XTICVL
C
C     DRAW TIC MARKS ON X-SCALE
C
```

```
779 -  783.000          DO 20 J= 3,12,3
780 -  784.000          TICMRK = TICMRK+XINTV
781 -  785.000          CALL MOVEA(XTIC(J),YSC(IPASSZ))
782 -  786.000          CALL DRAWA(XTIC(J),YTIC)
783 -  787.000          XTIC(J+1) = XTIC(J) - XTICX
784 -  788.000          ENCODE(4,802,AMEN)  TICMRK
785 -  789.000          IF(TICMRK .LE. 9.) ENCODE(4,800,AMEN)TICMRK
786 -  790.000          IF(TICMRK .GT. 9. .AND. TICMRK .LE. 59.)
787 -  791.000         &   ENCODE(4,801,AMEN)TICMRK
788 -  792.000          IF(TICMRK .GT. 99. .AND. TICMRK .LE. 999.)
789 -  793.000         &   ENCODE(4,802,AMEN)TICMRK
790 -  794.000      900 FORMAT(I1)
791 -  795.000      901 FORMAT(I3)
792 -  796.000      800 FORMAT(F3.0)
793 -  797.000      801 FORMAT(F4.1)
794 -  798.000      802 FORMAT(F4.1)
795 -  799.000          CALL KAM2AS(4,AMEN,AMENX)
796 -  800.000          CALL MOVEA(XTIC(J+1),YTIC3)
797 -  801.000          CALL HLABEL(4,AMENX)
798 -  802.000          XTIC(J+3) = XTIC(J)+XTICVL
799 -  803.000       20 CONTINUE
800 -  804.000 C
801 -  805.000 C       LABEL THE PLOT
802 -  806.000 C
803 -  807.000          YLABEL = YSC(IPASSZ) - .45
804 -  808.000          XLABEL = XTIC(3) - 2.
805 -  809.000          YLABEL = YSC(IPASSZ) - .389
806 -  810.000             RTITLE(:6)=IRTITL(1)
807 -  811.000             RTITLE(7:12)=IRTITL(2)
808 -  812.000             RTITLE(13:18)=IRTITL(3)
809 -  813.000             RTITLE(19:24)=IRTITL(4)
810 -  814.000             ENCODE(24,11,RTITLX)RTITLE
811 -  815.000       11    FORMAT(A24)
812 -  816.000             ENCODE(8,12,IIUNIX)IIUNIT
813 -  817.000       12    FORMAT(A6)
814 -  818.000             CALL KAM2AS(24,RTITLX,RTILX)
815 -  819.000             CALL KAM2AS(6,IIUNIX,IUNIX)
816 -  820.000             CALL MOVEA(XLABEL,YLABEL)
817 -  821.000             CALL HLABEL(24,RTILX)
818 -  822.000             CALL MOVEA(7.58,YLABEL)
819 -  823.000             CALL HLABEL(6,IUNIX)
820 -  824.000 C
821 -  825.000 C       DRAW TIC MARKS ON Y-SCALE
822 -  826.000 C
823 -  827.000      909 CONTINUE
824 -  828.000          TICYX= YSC(4)/2
825 -  829.000          TICY(1) = YSC(IPASSZ)+TICYX
826 -  830.000          TICY(2) = YSC(IPASSZ) + YSC(4)
827 -  831.000          TICX = XSC(1) - XTICX
828 -  832.000          TICX2 = TICX - (2*XTICX)
829 -  833.000 C
830 -  834.000          CALL MOVEA (XSC(1),TICY(1))
```

```
831 -  835.000        CALL DRAWA(TICX,TICY(1))
832 -  836.000        ENCODE(4,901,ANMEM)  INDMEM
833 -  837.000        IF(INDMEM .LE. 9)  ENCODE(4,900,ANMEM)INDMEM
834 -  838.000 C
835 -  839.000           CALL KAM2AS(3,ANMEM,ANMEMX)
836 -  840.000        CALL MOVEA(TICX2,TICY(1))
837 -  841.000        CALL HLABEL(3,ANMEMX)
838 -  842.000        ENCODE(4,901,ANMEM)  INMEM
839 -  843.000        IF (INMEM .LE. 9)  ENCODE(4,900,ANMEM)INMEM
840 -  844.000           CALL KAM2AS(3,ANMEM,ANMEMX)
841 -  845.000        CALL MOVEA(XSC(1),TICY(2))
842 -  846.000        CALL DRAWA(TICX,TICY(2))
843 -  847.000        CALL MOVEA(TICX2,TICY(2))
844 -  848.000        CALL HLABEL(3,ANMEMX)
845 -  849.000 C
846 -  850.000 C      DRAW DASHED LINE AT INITIAL VALUE LEVEL
847 -  851.000        DAINIT=(YFACT*YINIT)+YSC(IPASSZ)
848 -  852.000        IF(YTIC2.LT.DAINIT)GO TO 910
849 -  853.000        CALL MOVEA(XSC(1),DAINIT)
850 -  854.000        CALL LINE(7434)
851 -  855.000        CALL DRAWA(XSC(2),DAINIT)
852 -  856.000 C
853 -  857.000 C      LABEL THE Y-AXIS
854 -  858.000 C
855 -  859.000    910 Y(1)=TICY(1)+.547
856 -  860.000        X1=.3246
857 -  861.000 C
858 -  862.000           DO 99 I=1,5
859 -  863.000           ENCODE(4,13,AMEX)AME(1)
860 -  864.000     13    FORMAT(A1)
861 -  865.000           CALL KAM2AS(1,AMEX,AMEXX)
862 -  866.000           CALL MOVEA(X1,Y(I))
863 -  867.000           CALL HLABEL(1,AMEXX)
864 -  868.000           Y(I+1) = Y(I) - .2
865 -  869.000     99    CONTINUE
866 -  870.000           CALL TSEND
867 -  871.000 C
868 -  872.000 C      DRAW PLOTS
869 -  873.000 C
870 -  874.000     77 CONTINUE
871 -  875.000        CALL MOVEA(XSC(1),YSC(IPASSZ))
872 -  876.000        XPLOT2 = XSC(2) - XSC(1)
873 -  877.000 C
874 -  878.000           JJ=0
875 -  879.000           DO 40 J=1,NLVL
876 -  880.000           K =(J*2)-1
877 -  881.000           XDATA=(TIMES(K)-STIM4)*XFACT
878 -  882.000           YDATA=TIMES(K + 1)*YFACT
879 -  883.000 C
880 -  884.000 C           DETERMINE IF THIS TIME IS WITHIN BORDERS IF SEGMENT SET
881 -  885.000           ISET=7
882 -  886.000 C           WRITE(1,750)TIMES(K),TIMES(K+1)
```

```
883 -  887.000   750      FORMAT(2X,F5.1,1X,F5.1)
884 -  888.000            IF(ISEG.EQ.1)CALL CKHIST(TIMES(K),TIMES(K+2),ISET,NLVL,J)
885 -  889.000 C          WRITE(1,751)TIMES(K),TIMES(K+1),ISET
886 -  890.000   751      FORMAT(2X,F5.1,1X,F5.1,1X,I1)
887 -  891.000            IF(ISET.EQ.1)GO TO 40
888 -  892.000            IF(ISET.EQ.0)GO TO 30
889 -  893.000            XDATA=(TIMES(K)-STIMM)*XFACT
890 -  894.000            YDATA=TIMES(K+1)*YFACT
891 -  895.000    30     XSCAL=XDATA+XSC(1)
892 -  896.000            JJ=JJ+1
893 -  897.000 C
894 -  898.000            YSCALE=YDATA + YSC(IPASSZ)
895 -  899.000            IF(JJ .GT. 1) GO TO 666
896 -  900.000            IF(TIMES(K).EQ.WORDRF)GO TO 752
897 -  901.000            CALL MOVEA(XSCAL,YSC(IPASSZ))
898 -  902.000            CALL DRAWA(XSCAL,YSCALE)
899 -  903.000            GO TO 753
900 -  904.000   752      CALL MOVEA(XSC(1),YSC(IPASSZ))
901 -  905.000            CALL DRAWA(XSC(1),YSCALE)
902 -  906.000   753      IF(NLVL .LE. 1) CALL DRAWA(XSC(2),YSCALE)
903 -  907.000            IF(NLVL .LE. 1) GO TO 555
904 -  908.000            YSCLAS = YSCALE
905 -  909.000            YSCALE = 0.0
906 -  910.000            GO TO 40
907 -  911.000   666      CONTINUE
908 -  912.000            CALL DRAWA(XSCAL,YSCLAS)
909 -  913.000            CALL DRAWA (XSCAL,YSCALE)
910 -  914.000            YSCLAS = YSCALE
911 -  915.000            YSCALE = 0.0
912 -  916.000    40      CONTINUE
913 -  917.000 C
914 -  918.000         IF(TIMES(K+1).GT.0)CALL DRAWA(XSC(2),YSCLAS)
915 -  919.000         IF(TIMES(K+1).EQ.0)CALL DRAWA(XSCAL,YSC(IPASSZ))
916 -  920.000   555 CONTINUE
917 -  921.000         IF(IPASSZ .EQ. 3 .OR. ICOUNT .EQ. 1) GO TO 699
918 -  922.000         IF(IPASSZ .GT. 1 .AND. ICOUNT .EQ. 2) GO TO 699
919 -  923.000         CALL TSEND
920 -  924.000         GO TO 9999
921 -  925.000   699 CONTINUE
922 -  926.000         CALL MOVEA(.5,.5)
923 -  927.000         CALL TSEND
924 -  928.000         CALL WAIT(ILINE,3,ICPY)
925 -  929.000         IPASSZ=0
926 -  930.000 C
927 -  931.000 C      NORMAL EXIT
928 -  932.000 C
929 -  933.000  9000 CONTINUE
930 -  934.000         IF(IPASSZ .NE. ICOUNT)GO TO 9999
931 -  935.000         CALL MOVEA(.5,.5)
932 -  936.000         CALL TSEND
933 -  937.000         CALL WAIT(ILINE,3,ICPY)
934 -  938.000         CALL TINPUT(IC)
```

```
535 -   939.000   9999 CONTINUE
536 -   940.000        RETURN
537 -   941.000        END
538 -   942.000           SUBROUTINE INIT
539 -   943.000 C
540 -   944.000 C  THIS SUBROUTINE INITIALIZES THE TERMINAL ROUTINES
541 -   945.000 C
542 -   946.000 C          TEKTRONIX INITIALIZATION
543 -   947.000     10     CONTINUE
544 -   948.000            CALL INITI(120)
545 -   949.000            CALL TERM(3,4096)
546 -   950.000            CALL CHRSIZ(3)
547 -   951.000            GO TO 999
548 -   952.000 C          EXIT
549 -   953.000    999     CONTINUE
550 -   954.000            RETURN
551 -   955.000 C
552 -   956.000            END
553 -   957.000            SUBROUTINE INPUT
554 -   958.000 C
555 -   959.000         DIMENSION I2(4),RDCD(20)
556 -   960.000         CHARACTER LBL*2,ATABL*1,I12*2,I2*6
557 -   961.000         INTEGER RDCD*4
558 -   962.000         INCLUDE MATRIX
559 -   963.000         INCLUDE ERRORS
560 -   964.000 C
561 -   965.000 C          READ A CARD AND DETERMINE TYPE
562 -   966.000            IMTSK=0
563 -   967.000            IMRSC=0
564 -   968.000            MRSCS=0
565 -   969.000            IZ = 0
566 -   970.000           IX=0
567 -   971.000           TES2=0.0
568 -   972.000            MTASKS=0
569 -   973.000            CALL ASK
570 -   974.000      1     CONTINUE
571 -   975.000            READ(5,101)(RDCD(IIX),IIX=1,20)
572 -   976.000    101     FORMAT(20A4)
573 -   977.000            DECODE(80,100,RDCD) LBL
574 -   978.000    100     FORMAT(A2)
575 -   979.000            IF(LBL .EQ. 'C ') GO TO 1
576 -   980.000            IF(LBL.EQ.'TA') GOTO 5
577 -   981.000            IF(LBL .EQ. 'EO') GO TO 999
578 -   982.000            GO TO (10,20),ITABL
579 -   983.000            GO TO 1
580 -   984.000 C
581 -   985.000 C          TABLE CARD
582 -   986.000      5     CONTINUE
583 -   987.000            DECODE(80,105,RDCD) ATABL
584 -   988.000    105     FORMAT(6X,A1)
585 -   989.000         IF(ATABL .EQ. 1HF) ITABL = 1
586 -   990.000         IF(ATABL .EQ. 1HR) ITABL = 2
```

```
 987 -  991.000           GO TO 1
 988 -  992.000 C
 989 -  993.000 C         TASK DESCRIPTION CARD
 990 -  994.000     10    CONTINUE
 991 -  995.000           ICCK=2
 992 -  996.000           DECODE(80,110,RDCD) IC
 993 -  997.000    110    FORMAT(I1)
 994 -  998.000           IF(IC .NE. 0) GO TO 15
 995 -  999.000 C
 996 - 1000.000 C               NEW DESCRIPTION CARD
 997 - 1001.000                 I = NTASKS + 1
 998 - 1002.000                 IF(I .GT. MAXTSK) IMTSK=IMTSK+1;IFATAL=1
 999 - 1003.000                 NTASKS = I
1000 - 1004.000                 DECODE(80,111,RDCD) TASK(I),(TSKLBL(I,J),J=1,4),
1001 - 1005.000       &         IPFLG(I),ISC(I),ISHIFT(I),DUR(I),
1002 - 1006.000       &         (LPRED(I,J),J=1,4),(LSR(I,J),J=1,3)
1003 - 1007.000    111          FORMAT(1X,A4,4A6,A1,A1,I1,F5.1,4A4,3A8)
1004 - 1008.000                 SDUR(I)=DUR(I)
1005 - 1009.000           ITESS='X'
1006 - 1010.000           IF(IPFLG(I).NE.ITESS)GO TO 13
1007 - 1011.000           IX=IX+1
1008 - 1012.000           DECODE(80,303,DUR(I))TES2
1009 - 1013.000    303    FORMAT(F6.1)
1010 - 1014.000           NTASKS=I-1
1011 - 1015.000     13     ITES='Z'
1012 - 1016.000                 IF(ISHIFT(I) .EQ. 0) ISHIFT(I) = 3
1013 - 1017.000                 IF(IPFLG(I) .EQ. ITES) IZ = IZ + 1
1014 - 1018.000           IF(IZ .GT. 1 .OR. IX .GT. 1) IFATAL = 1
1015 - 1019.000           IF(IZ.EQ.1 .AND. IX.EQ.1) IFATAL=1
1016 - 1020.000                 GO TO 1
1017 - 1021.000 C
1018 - 1022.000 C               TASK CONTINUATION CARD
1019 - 1023.000     15          CONTINUE
1020 - 1024.000                 IF(IC.EQ.1)DECODE(80,115,RDCD) (LPRED(I,K),K=5,8),
1021 - 1025.000       1(LSR(I,J),J=4,6)
1022 - 1026.000                 IF(IC.EQ.2)DECODE(80,115,RDCD) (LPRED(I,K),K=9,12),
1023 - 1027.000       *(LSR(I,J),J=7,9)
1024 - 1028.000                 IF(IC.EQ.3)DECODE(80,116,RDCD)
1025 - 1029.000       *(LSR(I,J),J=10,12)
1026 - 1030.000    115          FORMAT(37X,4A4,3A8)
1027 - 1031.000    116          FORMAT(53X,3A8)
1028 - 1032.000                 GO TO 1
1029 - 1033.000 C
1030 - 1034.000 C         RESOURCE DESCRIPTION CARD
1031 - 1035.000     20    CONTINUE
1032 - 1036.000           I = NRSCS + 1
1033 - 1037.000           IF(I .GT. MAXRSC) IFATAL=1;IMRSC=IMRSC+1
1034 - 1038.000           NRSCS = I
1035 - 1039.000           DECODE(80,120,RDCD) RSC(I),(RSCLBL(I,J),J=1,4),
1036 - 1040.000       &   SMCK(I),VTYP(I),IFCN(I),VINIT(I),CRESF(I),CRESL(I)
1037 - 1041.000    120    FORMAT(1X,A2,1X,4A6,I1,I1,I1,F5.1,I1,F5.1)
1038 - 1042.000           IF(CRESF(I).EQ.0)CRESF(I)=2
```

```
1039 - 1043.000            GO TO 1
1040 - 1044.000  C
1041 - 1045.000  C         EXIT
1042 - 1046.000     999    CONTINUE
1043 - 1047.000            RETURN
1044 - 1048.000  C
1045 - 1049.000            END
1046 - 1050.000            SUBROUTINE INPUT2
1047 - 1051.000            DIMENSION RDCD(25)
1048 - 1052.000            INTEGER RDCD*4
1049 - 1053.000            INCLUDE MATRIX
1050 - 1054.000            INCLUDE ERRORS
1051 - 1055.000  C
1052 - 1056.000            INTSK=0
1053 - 1057.000            INRSC=0
1054 - 1058.000            IX=0
1055 - 1059.000            IZ=0
1056 - 1060.000            NRSCS=0
1057 - 1061.000            NTASKS=0
1058 - 1062.000            TES2=0.0
1059 - 1063.000  C
1060 - 1064.000  C         READ LENGTH OF DEFINITION BLOCKS
1061 - 1065.000            READ(5,101)(RDCD(IIX),IIX=1,5)
1062 - 1066.000     101    FORMAT(5A4)
1063 - 1067.000            DECODE(20,100,RDCD)LPDB,LRDB,LNDB,LRAD,LEDB
1064 - 1068.000     100    FORMAT(5I4)
1065 - 1069.000            READ(5,102)RDCD
1066 - 1070.000     102    FORMAT(A4)
1067 - 1071.000            DECODE(4,103,RDCD)RDBP
1068 - 1072.000     103    FORMAT(I4)
1069 - 1073.000  C
1070 - 1074.000  C         READ PDB BLOCK
1071 - 1075.000            READ(5,104)FTITLE
1072 - 1076.000     104    FORMAT(9A4)
1073 - 1077.000            READ(5,105)IDFTI
1074 - 1078.000     105    FORMAT(A4)
1075 - 1079.000            READ(5,106)ITUNIT
1076 - 1080.000     106    FORMAT(A4)
1077 - 1081.000            READ(5,107)LVFTI
1078 - 1082.000     107    FORMAT(A4)
1079 - 1083.000  C
1080 - 1084.000  C         READ RDB BLOCK
1081 - 1085.000            READ(5,200)NDBP
1082 - 1086.000     200    FORMAT(A4)
1083 - 1087.000            READ(5,202)RDCD
1084 - 1088.000     202    FORMAT(A4)
1085 - 1089.000            DECODE(4,203,RDCD)NRDB
1086 - 1090.000     203    FORMAT(I4)
1087 - 1091.000            IF(NRDB.EQ.0)GO TO 251
1088 - 1092.000            DO 250 I=1,NRDB
1089 - 1093.000                 READ(5,210)RSC(I)
1090 - 1094.000     210         FORMAT(A2)
```

```
1091 - 1095.000                READ(5,211)(RSCLBL(I,J),J=1,4)
1092 - 1096.000    211         FORMAT(6A4)
1093 - 1097.000                READ(5,212)RDCD
1094 - 1098.000    212         FORMAT(A4)
1095 - 1099.000                DECODE(4,213,RDCD)SMCK(I),TTYP(I),CRESF(I),CRESL(I)
1096 - 1100.000    213         FORMAT(4I1)
1097 - 1101.000                IF(CRESF(I).EQ.0)CRESF(I)=2
1098 - 1102.000                READ(5,214)RDCD
1099 - 1103.000    214         FORMAT(3A4)
1100 - 1104.000                DECODE(12,215,RDCD)VINIT(1)
1101 - 1105.000    215         FORMAT(F12.0)
1102 - 1106.000                READ(5,216)PDCD
1103 - 1107.000    216         FORMAT(A4)
1104 - 1108.000                DECODE(4,217,RDCD)IFCN(I)
1105 - 1109.000    217         FORMAT(I2)
1106 - 1110.000                READ(5,218)XNRDB
1107 - 1111.000    218         FORMAT(A4)
1108 - 1112.000    250     CONTINUE
1109 - 1113.000                IF(I.GT.MAXRSC)IFATAL=1;IMRSC=IMRSC+1
1110 - 1114.000                NRSCS=I
1111 - 1115.000    251     CONTINUE
1112 - 1116.000 C
1113 - 1117.000 C              READ NDB BLOCK
1114 - 1118.000                I=1
1115 - 1119.000                READ(5,300)EDBP
1116 - 1120.000    300     FORMAT(A4)
1117 - 1121.000    321     CONTINUE
1118 - 1122.000                READ(5,301)RDCD
1119 - 1123.000    301     FORMAT(A4)
1120 - 1124.000                DECODE(4,302,RDCD)NDBP
1121 - 1125.000    302     FORMAT(I4)
1122 - 1126.000    310     CONTINUE
1123 - 1127.000                READ(5,311)TASK(I)
1124 - 1128.000    311     FORMAT(A4)
1125 - 1129.000                READ(5,312)(TSKLBL(I,J),J=1,4)
1126 - 1130.000    312     FORMAT(6A4)
1127 - 1131.000                READ(5,311)RDCD
1128 - 1132.000                DECODE(4,314,RDCD)ISHIFT(I)
1129 - 1133.000    314         FORMAT(I1)
1130 - 1134.000                READ(5,311)RDCD
1131 - 1135.000                DECODE(4,316,RDCD)IPFLG(I),ISC(I)
1132 - 1136.000    316         FORMAT(2A1)
1133 - 1137.000                READ(5,317)RDCD
1134 - 1138.000    317         FORMAT(2A4)
1135 - 1139.000                DECODE(8,318,PDCD)DUR(I)
1136 - 1140.000    318         FORMAT(F5.1)
1137 - 1141.000                ITESS='Z'
1138 - 1142.000                IF(IPFLG(I).NE.ITESS)GO TO 13
1139 - 1143.000                IX=IX+1
1140 - 1144.000                DECODE(8,315,DUR(I))TES2
1141 - 1145.000    315         FORMAT(F6.1)
1142 - 1146.000     13         ITES='Z'
```

```
1145 - 1147.000           IF(ISHIFT(I).EQ.0)ISHIFT(I)=3
1146 - 1148.000           IF(IPFLG(I).EQ.ITES)IZ=IZ+1
1147 - 1149.000           IF(IZ.GT.1.OR.IX.GT.1)IFATAL=1
1148 - 1150.000           IF(IZ.EQ.1.AND.IX.EQ.1)IFATAL=1
1149 - 1151.000 C
1150 - 1152.000 C         DETERMINE IF RESOURCES FOR THIS NODE
1151 - 1153.000           II=1
1152 - 1154.000    325    CONTINUE
1153 - 1155.000           READ(5,319)RDCD
1154 - 1156.000    319    FORMAT(A4)
1155 - 1157.000           DECODE(4,320,RDCD)NRAD
1156 - 1158.000    320    FORMAT(I4)
1157 - 1159.000           READ(5,326)ICODE
1158 - 1160.000    326    FORMAT(A2)
1159 - 1161.000           READ(5,327)IC
1160 - 1162.000    327    FORMAT(A1)
1161 - 1163.000           READ(5,328)IQ
1162 - 1164.000    328    FORMAT(2A4)
1163 - 1165.000           ENCODE(12,329,LSR(I,II))IC,ICODE,IQ
1164 - 1165.500    329    FORMAT(A1,A2,2A4)
1165 - 1166.000           IF(NRAD.EQ.1)GO TO 350
1166 - 1167.000           II=II+1
1167 - 1168.000           GO TO 325
1168 - 1169.000    350    CONTINUE
1169 - 1170.000           IF(NDBP.EQ.0)GO TO 375
1170 - 1171.000           I=I+1
1171 - 1172.000           GO TO 321
1172 - 1173.000    375    CONTINUE
1173 - 1174.000           NTASKS=I
1174 - 1175.000 C
1175 - 1176.000 C         READ EDB BLOCK
1176 - 1177.000           READ(5,400)RDCD
1177 - 1178.000    400    FORMAT(A4)
1178 - 1179.000           DECODE(4,401,RDCD)NEDB
1179 - 1180.000    401    FORMAT(I4)
1180 - 1181.000           DO 475 I=1,NEDB
1181 - 1182.000               READ(5,402)EDGID
1182 - 1183.000    402        FORMAT(A4)
1183 - 1184.000               READ(5,402)EDGPR
1184 - 1185.000               READ(5,402)EDGSU
1185 - 1186.000               CALL EDGSET(EDGPR,EDGSU)
1186 - 1187.000    475    CONTINUE
1187 - 1188.000 C
1188 - 1189.000           RETURN
1189 - 1190.000           END
1190 - 1191.000           SUBROUTINE EDGSET(EDGPR,EDGSU)
1191 - 1191.500           INCLUDE PATRIX
1192 - 1192.000 C
1193 - 1193.000 C         INSERT PREDECESSOR WITH CORRECT NODE
1194 - 1194.000           DO 100 I=1,NTASKS
1195 - 1195.000               IF(TASK(I).NE.EDGSU)GO TO 100
1196 - 1196.000               DO 50 J=1,MAXLNK
```

```
1195 - 1197.000                     IF(LPRED(I,J).NE.BLANK.AND.J.EQ.MAXLNK)INTSK=INTSK+1;
1196 - 1198.000                     IFATAL=1;GO TO 150
1197 - 1199.000                     IF(LPRED(I,J).NE.BLANK)GO TO 50
1198 - 1200.000                     LPRED(I,J)=EDGPR
1199 - 1201.000                     GO TO 150
1200 - 1202.000                  50 CONTINUE
1201 - 1203.000                 100 CONTINUE
1202 - 1204.000                 150 CONTINUE
1203 - 1205.000 C
1204 - 1206.000                     RETURN
1205 - 1207.000                     END
1206 - 1208.000                     SUBROUTINE INSERT(NODE,LKSUCR,LIST)
1207 - 1209.000 C
1208 - 1210.000                     DIMENSION LIST(1)
1209 - 1211.000                     INCLUDE MATRIX
1210 - 1212.000 C
1211 - 1213.000 C                   GET LINK TO PREDICESSOR OF NEW NODE
1212 - 1214.000                     LKPRED = IPALFL(LIST(LKSUCR))
1213 - 1215.000 C
1214 - 1216.000 C                   SET LINKS IN NEW NODE
1215 - 1217.000                     LIST(NODE) = ISTRR(LKSUCR,LIST(NODE))
1216 - 1218.000                     LIST(NODE) = ISTRL(LKPRED,LIST(NODE))
1217 - 1219.000 C
1218 - 1220.000 C                   RESET SUCCESSOR LINK IN PREDICESSOR
1219 - 1221.000                     LIST(LKPRED) = ISTRR(NODE,LIST(LKPRED))
1220 - 1222.000 C
1221 - 1223.000 C                   RESET PREDICESSOR LINK IN SUCCESSOR
1222 - 1224.000                     LIST(LKSUCR) = ISTRL(NODE,LIST(LKSUCR))
1223 - 1225.000 C
1224 - 1226.000 C                   EXIT
1225 - 1227.000                 999 CONTINUE
1226 - 1228.000                     RETURN
1227 - 1229.000 C
1228 - 1230.000                     END
1229 - 1231.000                     SUBROUTINE PLOTBR(MARK,EARLY,LATE)
1230 - 1232.000 C
1231 - 1233.000                     PARAMETER MAXTSK = 300
1232 - 1234.000                     COMMON/QSEG/IGEGF
1233 - 1235.000                     COMMON/FLAG/IZB
1234 - 1236.000                     INCLUDE TPLOTS
1235 - 1237.000                     INTEGER EARLY,DNAMLX(6),DNAMX(24),XDUR(2),XDURX(8)
1236 - 1238.000                     CHARACTER*24 DNAME
1237 - 1239.000                     DIMENSION YS(19)
1238 - 1240.000                     DATA XSTRT/2.9275/
1239 - 1241.000 C
1240 - 1242.000 C                   SET UP COORDINATES
1241 - 1243.000                     DNAME(:6)=NAME(1)
1242 - 1244.000                     DNAME(7:12)=NAME(2)
1243 - 1245.000                     DNAME(13:18)=NAME(3)
1244 - 1246.000                     DNAME(19:24)=NAME(4)
1245 - 1247.000                     YL=ILINE
1246 - 1248.000                     YPT=(YL*.25)+.65
```

```
1247 - 1249.000           TX1=STRTX-STIM(ISEGF)
1248 - 1250.000           TX2=TX1+DURX
1249 - 1251.000           XS1=XSTRT+((TX1/TOT(ISEGF))*XSIM)
1250 - 1252.000           XS2=XSTRT+((TX2/TOT(ISEGF))*XSIM)
1251 - 1253.000           DRXS2=14.5-XS2
1252 - 1254.000           IF(IPASSX.NE.1)GO TO 15
1253 - 1255.000 C
1254 - 1256.000 C         PRINT NAME
1255 - 1257.000           YPT1=YPT+.05
1256 - 1258.000           IF(IZB.EQ.0)GO TO 10
1257 - 1259.000           ENCODE(24,50,DNAMEX)DNAME
1258 - 1260.000     50    FORMAT(A24)
1259 - 1261.000           CALL KAN2AS(24,DNAMEX,DNAMX)
1260 - 1262.000           CALL MOVEA(.32,YPT1)
1261 - 1263.000           CALL HLABEL(24,DNAMX)
1262 - 1264.000 C
1263 - 1265.000 C         DETERMINE WHERE DURATION IS TO BE PRINTED
1264 - 1266.000           DRXS1=XS1-2.9275
1265 - 1267.000           DRXS2=14.5-XS2
1266 - 1268.000           XXS1=XS1
1267 - 1269.000           IF(MARK.EQ.LATE)GO TO 1
1268 - 1270.000           DDUR=DURX
1269 - 1271.000      1    IF(IPASSX.NE.IPASS)GO TO 16
1270 - 1272.000     15    IF(MARK.EQ.LATE)GO TO 16
1271 - 1273.000           ENCODE(9,5,XDUR)DDUR
1272 - 1274.000           GO TO 6
1273 - 1275.000     16    ENCODE(8,5,XDUR)DURX
1274 - 1276.000      5    FORMAT(F7.1)
1275 - 1277.000      6    IF(DRXS1.GE.DRXS2)GO TO 30
1276 - 1278.000           IF(DRXS2.GT.DRXS1)GO TO 40
1277 - 1279.000 C
1278 - 1280.000 C         PRINT DURATION TO LEFT OF BAR
1279 - 1281.000     30    X=XXS1-(8*.12)
1280 - 1282.000           CALL KAN2AS(7,XDUR,XDURX)
1281 - 1283.000           CALL MOVEA(X,YPT1)
1282 - 1284.000           CALL HLABEL(7,XDURX)
1283 - 1285.000           GO TO 10
1284 - 1286.000 C
1285 - 1287.000 C         PRINT DURATION TO RIGHT OF BAR
1286 - 1288.000     40    X=XS2+.12
1287 - 1289.000           CALL KAN2AS(7,XDUR,XDURX)
1288 - 1290.000           CALL MOVEA(X,YPT1)
1289 - 1291.000           CALL HLABEL(7,XDURX)
1290 - 1292.000 C
1291 - 1293.000 C         PLOT BAR
1292 - 1294.000     10    YS1=YPT+.2
1293 - 1295.000           YS2=YPT+.05
1294 - 1296.000           CALL MOVEA(XS1,YS2)
1295 - 1297.000           CALL DRAWA(XS1,YS1)
1296 - 1298.000           CALL DRAWA(XS2,YS1)
1297 - 1299.000           CALL DRAWA(XS2,YS2)
1298 - 1300.000           CALL DRAWA(XS1,YS2)
```

```
C
C       CROSS-HATCH OPTION
        IF(IFLGX .NE. 2) GO TO 20
        XS11=(XS1+XS2)/2.
        YS11=(YS1+YS2)/2.
        CALL MOVEA(XS1,YS11)
        CALL DRAWA(XS11,YS1)
        CALL MOVEA(XS1,YS2)
        CALL DRAWA(XS2,YS1)
        CALL MOVEA(XS11,YS2)
        CALL DRAWA(XS2,YS11)
C
C       SOLID BAR OPTION
20      CONTINUE
        IF(IFLGX .NE. 3) GO TO 999
        DCT = (YS1 - YS2)/10
        YS(1) = YS 2 + DCT
        DO 22 I = 2,10
        YS(I) = YS(I-1)+DCT
22      CONTINUE
        DO 23 I = 1,10
        CALL MOVEA(XS1,YS(I))
        CALL DRAWA(XS2,YS(I))
23      CONTINUE
C
C       EXIT
999 CONTINUE
        RETURN
        END
        SUBROUTINE PRNTS(MM,IPCESS)
C
        INCLUDE MATRIX
        INCLUDE CCP
C
C       DETERMINE IF DATA HAS BEEN PROCESSED
        IF(IPCESS.EQ.0)PRINT 6;CALL WAIT(1,0,0);GO TO 999
6       FORMAT(' DATA HAS NOT BEEN PROCESSED, HIT RETURN '/
     *   ' AND CHOOSE A ''3'' ON THE MENU')
C
C       DETERMINE IF THERE ARE ERRORS IN INPUT
        IF(IFATAL.EQ.1)PRINT 5;CALL WAIT(1,0,0);GO TO 999
5       FORMAT(' THERE ARE ERRORS IN INPUT, HIT RETURN AND '/
     *   ' CHOOSE A ''6'' IN THE MENU')
C
C       DETERMINE WHICH TABLE YOU WOULD LIKE TO PRINT OUT
        IF(MM.NE.9.AND.MM.NE.16.AND.MM.NE.17) GO TO 300
C
C       PRINT PREDECESSOR-SUCCESSOR TABLES
C
C       MAKE SURE IF SEGMENT FLAG IS TURNED ON INFORMATION IS GIVEN
        IF(ISEG.EQ.1.AND.IORDRF.EQ.BLANK)WRITE(1,7))
     *  CALL WAIT(1,0,0);GO TO 999
```

```
1351 - 1353.000      7    FORMAT(' BEGINNING AND ENDING TASK NUMBERS WERE NOT GIVEN',
1352 - 1354.000       *     ' AND SEGMENT FLAG IS TURNED ON')
1353 - 1355.000           CALL CLEAR(3)
1354 - 1356.000           PRINT 100, (FTITLE(I),I=1,6)
1355 - 1357.000  100      FORMAT(//T37,6A6/T37,'PREDECESSOR-SUCCESSOR TABLE'//
1356 - 1358.000       &         ' SEQ  TASK',T49,'NO. ',7(' PRED'),'   NO. ',7(' SUCR')//
1357 - 1359.000       &       ' NBR  LABEL  TASK TITLE',T39,'PRED    1    2    3    4    5
1358 - 1360.000       &    6    7   SUCR   1    2    3    4    5    6    7'//)
1359 - 1361.000           LINE = 0
1360 - 1362.000           IOUTS=0
1361 - 1363.000           DO 10 I = 1,NTASKS
1362 - 1364.000           IF(ISEG.EQ.1)CALL CKTSKN(TASK(I),IQT,IOUTS,IORDRF,IORDRL)
1363 - 1365.000           IF(IQT.EQ.1)GO TO 10
1364 - 1366.000           PRINT 101, I,TASK(I),(TSKLBL(I,J),J=1,4),NPRED(I),(LPRED(I,J),
1365 - 1367.000       &         J=1,7),NSUCR(I),(LSUCR(I,J),J=1,7)
1366 - 1368.000   101     FORMAT(I4,2X,A4,3X,4A6,I4,2X,7(1X,A4),1X,I5,1X,7(1X,A4))
1367 - 1369.000           LINE = LINE + 1
1368 - 1370.000           IF(LINE .LT. 50) GO TO 10
1369 - 1371.000                CALL WAIT(LINE,3,ICPY)
1370 - 1372.000                PRINT 100, (FTITLE(II),II=1,6)
1371 - 1373.000     10    CONTINUE
1372 - 1374.000           CALL WAIT(LINE,3,ICPY)
1373 - 1375.000   300     IF(MM.NE.10.AND.MM.NE.16.AND.MM.NE.17)GO TO 30
1374 - 1376.000 C
1375 - 1377.000 C         MAKE SURE IF SEGMENT FLAG IS TURNED ON INFORMATION IS GIVEN
1376 - 1378.000           IF(ISEG.EQ.1.AND.WORDRL.EQ.0.0)WRITE(1,8);
1377 - 1379.000       *   CALL WAIT(I,3,0);GO TO 999
1378 - 1380.000      8    FORMAT(' BEGINNING AND ENDING START TIME WERE NOT GIVEN',
1379 - 1381.000       *     ' AND SEGMENT FLAG IS TURNED ON')
1380 - 1382.000 C
1381 - 1383.000 C         PRINT SCHEDULE TIMES TABLE
1382 - 1384.000           PRINT 120, (FTITLE(I),I=1,6)
1383 - 1385.000   120     FORMAT(//T34,6A6/T34,'TASK SCHEDULING TABLE'//
1384 - 1386.000       &         ' SEQ  TASK',T69,'SCALED',T80,'SLACK',T90,
1385 - 1387.000       &         'START',T102,'STOP'/' NBR  LABEL  TASK TITLE',T41,'DUR',
1386 - 1388.000       &              ',       SHIFT        DUR  ',3('    TIME   ')//)
1387 - 1389.000           I = HEAD
1388 - 1390.000     20    CONTINUE
1389 - 1391.000           I = IHALFR(NC(I))
1390 - 1392.000           IF(I .EQ. NULL) CALL WAIT(IQX,3,ICPY);GO TO 30
1391 - 1393.000           TSTOP = STIME(I) + SDUR(I)
1392 - 1394.000           IF(ISEG.EQ.1)CALL CKSTTM(STIME(I),IQT,TSTOP,WORDRF,WORDRL)
1393 - 1395.000           IF(IQT.EQ.1)GO TO 20
1394 - 1396.000           PRINT 121, I, TASK(I),(TSKLBL(I,J),J=1,4),DUR(I),
1395 - 1397.000       +         ISHIFT(I),SDUR(I),SLACK(I),STIME(I),TSTOP
1396 - 1398.000   121     FORMAT(I4,2X,A4,3X,4A6,F7.0,11X,I6,3X,4F10.1)
1397 - 1399.000           LINE = LINE + 1
1398 - 1400.000           IF(LINE .LT. 50) GO TO 20
1399 - 1401.000                CALL WAIT(LINE,3,ICPY)
1400 - 1402.000                PRINT 120, (FTITLE(II),II=1,6)
1401 - 1403.000                GO TO 20
1402 - 1404.000 C
```

```
1403 - 1405.000 C         PRINT RESOURCE HISTOGRAMS
1404 - 1406.000     30    CONTINUE
1405 - 1407.000           IF(MM.NE.11.AND.MM.NE.16.AND.MM.NE.17)GO TO 310
1406 - 1408.000 C
1407 - 1409.000 C         MAKE SURE IF SEGMENT FLAG IS TURNED ON INFORMATION IS GIVEN
1408 - 1410.000           IF(ISEG.EQ.1.AND.WORDRL.EQ.0.0)WRITE(1,8);
1409 - 1411.000       *   CALL WAIT(I,0,9);GO TO 999
1410 - 1412.000           CALL CLEAR(4)
1411 - 1413.000           INDX = 1
1412 - 1414.000     31    CONTINUE
1413 - 1415.000           IRSC = HTABLE(INDX)
1414 - 1416.000         IF(IRSC .EQ. 0) GO TO 999
1415 - 1417.000           PRINT 130, (RSCLBL(IRSC,I),I=1,4), ITUNIT,HTABLE(INDX+1),
1416 - 1418.000      &    VINIT(IRSC),CNRS(CRESF(IPSC))
1417 - 1419.000    130    FORMAT(//39X,'RESOURCE HISTOGRAM FOR ',4A6/
1418 - 1420.000      &         37X,'TIME UNITS: ',A6,5X,'MAXIMUM LEVEL:',F7.0/
1419 - 1421.000      &         32X,'INITIAL VALUE: ',F5.1,5X,'CONSTRAINED RESOURCE: ',A3/
1420 - 1422.000      &         6(5X,'TIME    LEVEL  ')/)
1421 - 1423.000           LINE = LINE + 6
1422 - 1424.000           IF(LINE .GE. 49) CALL WAIT(LINE,4,ICPY)
1423 - 1425.000           NLVL = HTABLE(INDX+2)
1424 - 1426.000           K = (NLVL+5)/6
1425 - 1427.000           IN3 = INDX + 3
1426 - 1428.000           IF(NLVL .EQ. 0) GO TO 36
1427 - 1429.000           DO 35 I = 1,K
1428 - 1430.000           IS = ((I*12) - 11) + IN3
1429 - 1431.000           IE = (I*12) + IN3
1430 - 1432.000           IF(I.EQ.1)ISS=IS
1431 - 1433.000           IF(I .EQ. K) IE = (NLVL - (K-1)*6)*2 + IS - 1
1432 - 1434.000     35    CONTINUE
1433 - 1435.000           IS=ISS
1434 - 1436.000           IF(ISEG.EQ.1)CALL CKHTAB(HTABLE,IS,IE,WORDRF,WORDRL)
1435 - 1437.000           ISIE=IE-IS
1436 - 1438.000           K=((ISIE/2)+5)/6
1437 - 1439.000           DO 9 I=1,K
1438 - 1440.000           IEE=IS+11
1439 - 1441.000           IF(I.EQ.K)IEE=IE
1440 - 1442.000           PRINT 131, (HTABLE(J),J=IS,IEE)
1441 - 1443.000    131    FORMAT(6(F10.1,F7.0,3X))
1442 - 1444.000           LINE = LINE + 1
1443 - 1445.000           IF( LINE .GE. 49) CALL WAIT(LINE,4,ICPY)
1444 - 1446.000           IS=IEE+1
1445 - 1447.000      9    CONTINUE
1446 - 1448.000     36    CONTINUE
1447 - 1449.000           INDX = HTABLE(IN3)
1448 - 1450.000           IF(INDX .NE. 0) GO TO 31
1449 - 1451.000 C
1450 - 1452.000 C         WAIT FOR CUE BEFORE CONTINING
1451 - 1453.000           CALL WAIT(LINE,4,ICPY)
1452 - 1454.000 C
1453 - 1455.000 C         PRINT FULL TITLE TABLE
1454 - 1456.000    310    IF(MM.NE.12.AND.MM.NE.15.AND.MM.NE.16)GO TO 999
```

```
1455 - 1457.000 C
1456 - 1458.000 C         MAKE SURE IF SEGMENT FLAG IS ON INFORMATION IS GIVEN
1457 - 1459.000           IF(ISEG.EQ.1.AND.IORDPF.EQ.BLANK)WRITE(1,7);
1458 - 1460.000       *   CALL WAIT(1,0,0);GO TO 999
1459 - 1461.000           CALL PRTRET
1460 - 1462.000 C
1461 - 1463.000 C         EXIT
1462 - 1464.000    999    CONTINUE
1463 - 1465.000           RETURN
1464 - 1466.000 C
1465 - 1467.000           END
1466 - 1468.000           SUBROUTINE RESRCE(PRNTNW)
1467 - 1469.000 C
1468 - 1470.000 C  THIS SUBROUTINE PERFORMS ERROR CHECKING ON THE
1469 - 1471.000 C  RESOURCE SEIZE RELEASE TABLE
1470 - 1472.000 C
1471 - 1473.000           INCLUDE MATRIX
1472 - 1474.000           INTEGER PRNTNW
1473 - 1475.000           DIMENSION XIQSR(1)
1474 - 1476.000           EQUIVALENCE (XIQSR(1),SC(1))
1475 - 1477.000           CHARACTER ICODE*2,IC*1,ISIGN(4)*1
1476 - 1478.000           DATA ISIGN/' ','*','+','-'/
1477 - 1479.000          ICODE = '  '
1478 - 1480.000           IC=' '
1479 - 1481.000 C
1480 - 1482.000 C         OUTER LOOP - STEP THROUGH TASK LIST
1481 - 1483.000           DO 40 IT = 1,NTASKS
1482 - 1484.000           NSR(IT)=0
1483 - 1485.000 C
1484 - 1486.000 C         MIDDLE LOOP - SCAN SEIZE-RELEASE LIST FOR EACH TASK
1485 - 1487.000           DO 40 JR = 1,MAXSR
1486 - 1488.000           ICODE=LSR(IT,JR)(7:8)
1487 - 1489.000           IF(ICODE .EQ. '  ') GO TO 40
1488 - 1490.000           NSR(IT) = NSR(IT) + 1
1489 - 1491.000 C
1490 - 1492.000 C         INNER LOOP - CHECK S/R CODE AGAINST RESOURSE TABLE
1491 - 1493.000           DO 10 K = 1,MAXRSC
1492 - 1494.000           IR = K
1493 - 1495.000           IF(ICODE .EQ. RSC(IR)) GO TO 30
1494 - 1496.000    10     CONTINUE
1495 - 1497.000 C
1496 - 1498.000 C              DIAGNOSTIC - RESOURCE CODE IS UNDEFINED
1497 - 1499.000                IF(PRNTNW.EQ.1)PRINT 900, ICODE, TASK(IT)
1498 - 1500.000    900         FORMAT(' WARNING - RESOURCE CODE ',A2,' IS UNDEFINED. ',
1499 - 1501.000       &             'REFERENCED BY TASK ',A4)
1500 - 1502.000                IFATAL = 1
1501 - 1503.000                IR = NRSCS + 1
1502 - 1504.000                IF(IR .GT. MAXRSC) GO TO 20
1503 - 1505.000                NRSCS = IR
1504 - 1506.000                IF(PRNTNW.EQ.1)RSC(IR) = ICODE
1505 - 1507.000                RSCLBL(IR,1) = 'UNDEFI'
1506 - 1508.000                RSCLBL(IR,2) = 'NED   '
```

```
1507 - 1509.000                RSCLBL(IR,3) = '       '
1508 - 1510.000                RSCLBL(IR,4) = '       '
1509 - 1511.000                GO TO 30
1510 - 1512.000 C
1511 - 1513.000 C              FATAL ERROR - TOO MANY RESOURCES
1512 - 1514.000     20         CONTINUE
1513 - 1515.000                IF(PRNTNW.EQ.1)WRITE(1,901)
1514 - 1516.000    901         FORMAT(' *** FATAL ERROR - NUMBER OF RESOURCES EXCEEDS ',
1515 - 1517.000       &             'AVAILABLE STORAGE ***')
1516 - 1518.000                IFATAL = 1
1517 - 1519.000                GO TO 40
1518 - 1520.000 C
1519 - 1521.000 C          CHECK SEIZE/RELEASE CODE
1520 - 1522.000     30     CONTINUE
1521 - 1523.000            IC=LSR(IT,JR)(:1)
1522 - 1524.000            IF(IC .EQ. ISIGN(3) .OR. IC .EQ. ISIGN(4)) GO TO 35
1523 - 1525.000            IF(IC.EQ.ISIGN(1).OR.IC.EQ.ISIGN(2)) XIQSR(IR)=0.
1524 - 1526.000            IF(IC.EQ.ISIGN(1).OR.IC.EQ.ISIGN(2))GO TO 40
1525 - 1527.000 C
1526 - 1528.000 C              FATAL ERROR - INVALID SEIZE RELEASE CODE
1527 - 1529.000                IF(PRNTNW.EQ.1)PRINT 903, LSR(IT,JR), TASK(IT)
1528 - 1530.000    903         FORMAT(' *** FATAL ERROR - INVALID SEIZE/RELEASE CODE ',
1529 - 1531.000       &             A5,' IN TASK ',A4,' ***')
1530 - 1532.000                IFATAL = 1
1531 - 1533.000                GO TO 40
1532 - 1534.000 C
1533 - 1535.000 C          COUNT SEIZE/RELEASES
1534 - 1536.000     35     CONTINUE
1535 - 1539.000            DECODE(80,135,LSR(IT,JR)) XIQ
1536 - 1540.000    135     FORMAT(1X,F5.0)
1537 - 1543.000            IF(VTYP(IR).EQ.1)GO TO 69
1538 - 1544.000            IQ=XIQ
1539 - 1545.000            IF(IQ .EQ. 0) IQ  = 1
1540 - 1546.000            IF(IC .EQ. ISIGN(4)) IQ = -IQ
1541 - 1547.000            XIQ=IQ
1542 - 1548.000            GO TO 61
1543 - 1549.000     60     CONTINUE
1544 - 1552.000            DECODE(80,136,LSR(IT,JR))XIQD
1545 - 1553.000    136     FORMAT(4X,F2.1)
1546 - 1556.000            XIQ=XIQ+XIQD
1547 - 1557.000            IF(XIQ.EQ.0.)XIQ=1.
1548 - 1558.000            IF(IC.EQ.ISIGN(4))XIQ=-XIQ
1549 - 1559.000     61     XIQSR(IR) = XIQSR(IR) + XIQ
1550 - 1560.000 C
1551 - 1561.000 C          END OF LOOPS
1552 - 1562.000     40     CONTINUE
1553 - 1563.000 C
1554 - 1564.000 C          CHECK SEIZE/RELEASE TOTALS
1555 - 1565.000            DO 50 IR = 1,NRSCS
1556 - 1568.000            IF(XIQSR(IR) .EQ. 0) GO TO 50
1557 - 1569.000 C
1558 - 1570.000 C              DIAGNOSTIC - NON ZERO SUM RESOURCE
```

```
1559 - 1573.000                IF(SMCK(IR).EQ.3)GO TO 50
1560 - 1574.000                IF(FRETNW.EQ.1)PRINT 904, RSC(IR), NIQSR(IR)
1561 - 1575.000    904         FORMAT(' WARNING -FLOW DOES NOT PROVIDE ZERO-SUM SEIZE/',
1562 - 1576.000       &             'RELEASES FOR RESOURCE ',A2,', NET =',F8.1)
1563 - 1577.000                IFATAL = 1
1564 - 1578.000 C
1565 - 1579.000     50     CONTINUE
1566 - 1580.000 C
1567 - 1581.000 C          EXIT
1568 - 1582.000    999     CONTINUE
1569 - 1583.000            RETURN
1570 - 1584.000 C
1571 - 1585.000            END
1572 - 1586.000            SUBROUTINE SCHED(IPASS)
1573 - 1587.000 C
1574 - 1588.000            INCLUDE PATRIX
1575 - 1589.000            DATA IZERO,ILT /'Z    ','L    '/
1576 - 1590.000            CHARACTER IZERO
1577 - 1591.000 C
1578 - 1592.000 C          INITIALIZE
1579 - 1593.000            SCPTR = HEAD
1580 - 1594.000            LSORT = HEAD
1581 - 1595.000            WC(HEAD) = ISWRR(HEAD,WC(HEAD))
1582 - 1596.000            WC(HEAD) = ISTRL(HEAD,WC(HEAD))
1583 - 1597.000            NSCHED = 0
1584 - 1598.000            STMIN = 0.
1585 - 1599.000 C
1586 - 1600.000 C          IF LAST JOB, EXIT
1587 - 1601.000     10     CONTINUE
1588 - 1602.000            LINK1 = IHALFR(SC(SCPTR))
1589 - 1603.000            IF(LINK1 .EQ. NULL) GO TO 70
1590 - 1604.000            SCPTR = LINK1
1591 - 1605.000 C
1592 - 1606.000 C          SEPARATE EARLY AND LATE MARKED TASKS
1593 - 1607.000            IF(MARK(SCPTR) .EQ. LATE) GO TO 50
1594 - 1608.000 C
1595 - 1609.000 C          SCHEDULE TASKS MARKED EARLY
1596 - 1610.000            IF(NSCHED .NE. 0) GO TO 15
1597 - 1611.000 C
1598 - 1612.000 C          PROCESS FIRST EARLY MARK
1599 - 1613.000            STIME(SCPTR) = 0.
1600 - 1614.000            ISCHED(SCPTR) = 1
1601 - 1615.000            NSCHED = NSCHED + 1
1602 - 1616.000            GO TO 10
1603 - 1617.000 C
1604 - 1618.000 C          PROCESS REMAINING EARLY MARKS
1605 - 1619.000     15     CONTINUE
1606 - 1620.000            ISFLG = 0
1607 - 1621.000            NP = NPRED(SCPTR)
1608 - 1622.000            DO 20 I = 1,NP
1609 - 1623.000            LP = PRED(SCPTR,I)
1610 - 1624.000            IF(ISCHED(LP) .EQ. 0) GO TO 20
```

```
1611 - 1625.000           ENDT = STIME(LP) + SDUR(LP)
1612 - 1626.000           IF(ENDT .LE. ENDTL .AND. ISFLG .EQ. 1) GO TO 20
1613 - 1627.000           ENDTL = ENDT
1614 - 1628.000           ISFLG = 1
1615 - 1629.000      20   CONTINUE
1616 - 1630.000           STIME(SCPTR) = ENDTL
1617 - 1631.000           ISCHED(SCPTR) = 1
1618 - 1632.000           NSCHED = NSCHED + 1
1619 - 1633.000           GO TO 10
1620 - 1634.000 C
1621 - 1635.000 C         SCHEDULE TASKS MARKED LATE
1622 - 1636.000      50   CONTINUE
1623 - 1637.000           IF(NSCHED .NE. 0) GO TO 51
1624 - 1638.000 C
1625 - 1639.000 C         PROCESS FIRST LATE MARK
1626 - 1640.000           STIME(SCPTR)=0.
1627 - 1641.000           ISCHED(SCPTR)=1
1628 - 1642.000           NSCHED = NSCHED + 1
1629 - 1643.000           GO TO 10
1630 - 1644.000 C
1631 - 1645.000 C         PROCESS REMAINING LATE MARKS
1632 - 1646.000      51   ISFLG = 0
1633 - 1647.000           NS = NSUCR(SCPTR)
1634 - 1648.000           DO 60 I = 1,NS
1635 - 1649.000           LES = SUCR(SCPTR,I)
1636 - 1650.000           IF(ISCHED(LES) .EQ. 0) GO TO 60
1637 - 1651.000           IF(STIME(LES) .GE. STESS .AND. ISFLG .EQ. 1) GO TO 60
1638 - 1652.000           STESS = STIME(LES)
1639 - 1653.000           ISFLG = 1
1640 - 1654.000      60   CONTINUE
1641 - 1655.000           STIME(SCPTR) = STESS - SDUR(SCPTR)
1642 - 1656.000           ISCHED(SCPTR) = 1
1643 - 1657.000           NSCHED = NSCHED + 1
1644 - 1658.000           IF(STIME(SCPTR) .LT. STMIN) STMIN = STIME(SCPTR)
1645 - 1659.000           GO TO 10
1646 - 1660.000 C
1647 - 1661.000 C         IF A TASK IS FLAGGED TO HAVE START TIME OF ZERO
1648 - 1662.000      70   CONTINUE
1649 - 1663.000           DO 71 I = 1,NTASKS
1650 - 1664.000                IF(IPFLG(I) .EQ. IZERO) STMIN = STIME(I)
1651 - 1665.000      71   CONTINUE
1652 - 1666.000 C
1653 - 1667.000 C         BIAS START TIMES
1654 - 1668.000           IF(STMIN .EQ. 0.) GO TO 80
1655 - 1669.000           DO 75 I = 1,NTASKS
1656 - 1670.000                STIME(I) = STIME(I) - STMIN
1657 - 1671.000      75   CONTINUE
1658 - 1672.000 C
1659 - 1673.000 C         COMPUTE SLACK TIME
1660 - 1674.000      80   CONTINUE
1661 - 1675.000           DO 81 I = 1,NTASKS
1662 - 1676.000                SLACK(I) = ABS( SLACK(I) - STIME(I) )
```

```
1663 - 1677.000     91    CONTINUE
1664 - 1678.000 C
1665 - 1679.000 C         DETERMINE IF TASK EARLY OR LATE DEPENDING ON FLAG
1666 - 1680.000           DO 90 I=1,NTASKS
1667 - 1681.000                IF(PARK(I).EQ.EARLY.AND.ISC(I).EQ.BLANK)GO TO 89
1668 - 1682.000                IF(PARK(I).EQ.LATE.AND.ISC(I).EQ.ILT)GO TO 89
1669 - 1683.000                IF(PARK(I).NE.LATE.OR.ISC(I).NE.BLANK)GO TO 82
1670 - 1684.000                     STIME(I)=STIME(I)-SLACK(I)
1671 - 1685.000                     PARK(I)=EARLY
1672 - 1686.000                     GO TO 89
1673 - 1687.000     82              IF(PARK(I).NE.EARLY.OR.ISC(I).NE.ILT)GO TO 89
1674 - 1688.000                     STIME(I)=STIME(I)+SLACK(I)
1675 - 1689.000                     PARK(I)=LATE
1676 - 1690.000     89              CALL SORT(I,LSORT,STIME,SDUR,WC,HEAD)
1677 - 1691.000     90    CONTINUE
1678 - 1692.000 C
1679 - 1693.000 C         EXIT
1680 - 1694.000    999    CONTINUE
1681 - 1695.000           RETURN
1682 - 1696.000 C
1683 - 1697.000           END
1684 - 1698.000           SUBROUTINE SEQNCE
1685 - 1699.000 C
1686 - 1700.000 C     THISSUBROUTINE COMPLETES THE PRECIDENCE-SUCCESSOR SEQUENCES
1687 - 1701.000 C
1688 - 1702.000           INCLUDE PATRIX
1689 - 1703.000           INCLUDE ERRORS
1690 - 1704.000 C
1691 - 1705.000 C         INITIALIZE LIST POINTERS
1692 - 1706.000           SC(HEAD) = ISTRL(HEAD,SC(HEAD))
1693 - 1707.000           WC(HEAD) = ISTRL(HEAD,WC(HEAD))
1694 - 1708.000           SC(HEAD) = ISTRR(HEAD,SC(HEAD))
1695 - 1709.000           WC(HEAD) = ISTRR(HEAD,WC(HEAD))
1696 - 1710.000 C
1697 - 1711.000 C         CONVERT PRECIDENCE LABELS TO SEQUENCE NUMBERS
1698 - 1712.000     20    CONTINUE
1699 - 1713.000           DO 30 IT = 1,NTASKS
1700 - 1714.000           NPRED(IT) = 0
1701 - 1715.000           ERTSK(IT)=0
1702 - 1716.000           ERRTSK(IT)=0
1703 - 1717.000 C
1704 - 1718.000 C              SCAN PREDECESSOR LABELS
1705 - 1719.000                DO 29 IP = 1,MAXLNK
1706 - 1720.000                IF(LPRED(IT,IP) .EQ. BLANK) GO TO 30
1707 - 1721.000 C
1708 - 1722.000 C                   FIND SEQUENCE NUMBER FOR LABEL
1709 - 1723.000                     DO 21 IS = 1,NTASKS
1710 - 1724.000                     IF(LPRED(IT,IP).EQ.TASK(IS)) GOTO 22
1711 - 1725.000     21              CONTINUE
1712 - 1726.000                     ERTSK(IT)=TASK(IT)
1713 - 1727.000                     ERLPD(IT,IP)=LPRED(IT,IP)
1714 - 1728.000                     IFATAL = 1
```

```
1715 - 1729.000                      GO TO 29
1716 - 1730.000 C
1717 - 1731.000 C                    REPLACE PREDECESSOR LABEL WITH SEQUENCE NUMBER
1718 - 1732.000      22              CONTINUE
1719 - 1733.000                      PRED(IT,IP) = IS
1720 - 1734.000                      NPRED(IT) = NPRED(IT) + 1
1721 - 1735.000      29          CONTINUE
1722 - 1736.000      30      CONTINUE
1723 - 1737.000 C
1724 - 1738.000 C           BUILD SUCCESSOR LINKAGES-BY WORKING BACK THROUGH PREDECESSOR LI . NKS
1725 - 1739.000             DO 34 I=1,NTASKS
1726 - 1740.000    34       NSUCR(I)=0
1727 - 1741.000             DO 40 IT = 1,NTASKS
1728 - 1742.000             IF(NPRED(IT) .EQ. 0) GO TO 40
1729 - 1743.000 C
1730 - 1744.000 C                 SCAN PREDECESSOR LINKS
1731 - 1745.000                   DO 35 IP = 1,MAXLNK
1732 - 1746.000                   IF(PRED(IT,IP) .EQ. BLANK) GO TO 40
1733 - 1747.000 C
1734 - 1748.000 C                      SET SUCCESSOR LINKS FOR THIS PREDECESSOR
1735 - 1749.000                        NP = PRED(IT,IP)
1736 - 1750.000                        NSUCR(NP) = NSUCR(NP) + 1
1737 - 1751.000                        IF(NSUCR(NP) .GT. MAXLNK) GO TO 31
1738 - 1752.000                        IS = NSUCR(NP)
1739 - 1753.000                        SUCR(NP,IS) = IT
1740 - 1754.000                        LSUCR(NP,IS) = TASK(IT)
1741 - 1755.000                        GO TO 35
1742 - 1756.000      31                CONTINUE
1743 - 1757.000                        ERRTSK(NP)=TASK(NP)
1744 - 1758.000                        IFATAL = 1
1745 - 1759.000                        GO TO 40
1746 - 1760.000      35            CONTINUE
1747 - 1761.000      40        CONTINUE
1748 - 1762.000 C
1749 - 1763.000 C             EXIT
1750 - 1764.000    999        CONTINUE
1751 - 1765.000               RETURN
1752 - 1766.000 C
1753 - 1767.000               END
1754 - 1768.000           SUBROUTINE SETUP(IK)
1755 - 1769.000 C
1756 - 1770.000           INCLUDE MATRIX
1757 - 1771.000           INCLUDE TPLOTS
1758 - 1772.000 C
1759 - 1773.000 C         THIS DOES THE SETUP FOR PLOTING A TIMELINE GRAPH
1760 - 1774.000           DO 10 I = 1,4
1761 - 1775.000    10     NAME(I) = TSKLBL(IK,I)
1762 - 1776.000           IF(SLACK(IK) .LT. -0.01 .OR. SLACK(IK) .GT. 0.01) GO TO 20
1763 - 1777.000 C
1764 - 1778.000 C         CRITICAL PATH ACTIVITY
1765 - 1779.000           IPASS = 1
1766 - 1780.000           IFLG(1)=3
```

```
1767 - 1781.000            IET="E"
1768 - 1782.000          IF(IPFLG(IK).EQ.IET)IFLG(3) = 4
1769 - 1783.000          STRT(1) = STIME(IK)
1770 - 1784.000          DURR(1) = SDUR(IK)
1771 - 1785.000          GO TO 30
1772 - 1786.000 C
1773 - 1787.000 C        JOB NOT CRITICAL
1774 - 1788.000    20    CONTINUE
1775 - 1789.000          IF(MARK(WCPTR).EQ.LATE)GO TO 40
1776 - 1790.000 C
1777 - 1791.000 C        EARLY SCHEDULED JOB
1778 - 1792.000          IPASS = 2
1779 - 1793.000          IFLG(1) = 2
1780 - 1794.000          STRT(1) = STIME(IK)
1781 - 1795.000          DURR(1) = SDUR(IK)
1782 - 1796.000          IFLG(2) = 1
1783 - 1797.000          STRT(2) = STRT(1) + DURR(1)
1784 - 1798.000          DURR(2) = SLACK(IK)
1785 - 1799.000          GO TO 30
1786 - 1800.000 C
1787 - 1801.000 C        LATE SCHEDULED JOB
1788 - 1802.000    40 IPASS=2
1789 - 1803.000          IFLG(2)=2
1790 - 1804.000          STRT(2)=STIME(IK)
1791 - 1805.000          DURR(2)=SDUR(IK)
1792 - 1806.000          IFLG(1)=1
1793 - 1807.000          DURR(1)=SLACK(IK)
1794 - 1808.000          STRT(1)=STRT(2)-DURR(1)
1795 - 1809.000 C
1796 - 1810.000    30    CONTINUE
1797 - 1811.000 C
1798 - 1812.000          RETURN
1799 - 1813.000 C
1800 - 1814.000          END
1801 - 1815.000             SUBROUTINE SORT(NODE,IPTR,VALUE,TIE,LIST,LHEAD)
1802 - 1816.000 C
1803 - 1817.000             DIMENSION VALUE(1),LIST(1),TIE(1)
1804 - 1818.000 C
1805 - 1819.000 C           INITIALIZE
1806 - 1820.000             VNODE = VALUE(NODE)
1807 - 1821.000             TNODE = TIE(NODE)
1808 - 1822.000             IF(IPTR .EQ. LHEAD .OR. VNODE .GT. VALUE(IPTR)) GO TO 20
1809 - 1823.000             IF(VNODE .EQ. VALUE(IPTR) .AND. TNODE .GE. TIE(IPTR)) GO TO 20
1810 - 1824.000 C
1811 - 1825.000 C           SEARCH BACKWARD
1812 - 1826.000     10      CONTINUE
1813 - 1827.000             IPTR = IHALFL(LIST(IPTR))
1814 - 1828.000             IF(IPTR .EQ. LHEAD) GO TO 20
1815 - 1829.000             IF(VNODE .LE. VALUE(IPTR)) GO TO 10
1816 - 1830.000 C
1817 - 1831.000 C           SEARCH FORWARD
1818 - 1832.000     20      CONTINUE
```

```
1819 - 1833.000           IPTR = IHALFR(LIST(IPTR))
1820 - 1834.000           IF(IPTR .EQ. LHEAD) GO TO 30
1821 - 1835.000           IF(VNODE .GT. VALUE(IPTR)) GO TO 20
1822 - 1836.000           IF(VNODE.EQ.VALUE(IPTR) .AND. TIE(NODE).GE.TIE(IPTR)) GO TO 20
1823 - 1837.000 C
1824 - 1838.000 C         INSERT NODE AS PREDICESSOR OF NODE AT IPTR
1825 - 1839.000     30    CONTINUE
1826 - 1840.000           CALL INSERT(NODE,IPTR,LIST)
1827 - 1841.000 C
1828 - 1842.000 C         SET IPTR TO NODE JUST INSERTED
1829 - 1843.000           IPTR = NODE
1830 - 1844.000 C
1831 - 1845.000 C         EXIT
1832 - 1846.000    999    CONTINUE
1833 - 1847.000           RETURN
1834 - 1848.000 C
1835 - 1849.000           END
1836 - 1850.000           SUBROUTINE TRNGLE
1837 - 1851.000 C
1838 - 1852.000           INCLUDE TPLOTS
1839 - 1853.000          COMMON/FLAG/IZB
1840 - 1854.000          COMMON/QSEG/ISEGF
1841 - 1855.000           CHARACTER*24 DNAME
1842 - 1856.000          INTEGER DNAMEX(6),DNAMX(24)
1843 - 1857.000          DATA XSTRT/2.9275/
1844 - 1858.000          YL=ILINE
1845 - 1859.000 C
1846 - 1860.000          YPT=(YL*.25)+.65
1847 - 1861.000 C
1848 - 1862.000 C         PRINT NAME
1849 - 1863.000 C
1850 - 1864.000           DNAME(:6)=NAME(1)
1851 - 1865.000           DNAME(7:12)=NAME(2)
1852 - 1866.000           DNAME(13:18)=NAME(3)
1853 - 1867.000           DNAME(19:24)=NAME(4)
1854 - 1868.000          YPT1=YPT+.05
1855 - 1869.000          YS1=YPT+.2
1856 - 1870.000          YS2=YPT+.05
1857 - 1871.000          YS3=YPT+.125
1858 - 1872.000          ENCODE(24,10,DNAMEX)DNAME
1859 - 1873.000     10    FORMAT(A24)
1860 - 1874.000          CALL KAN2AS(24,DNAMEX,DNAMX)
1861 - 1875.000          CALL MOVEA(.02,YPT1)
1862 - 1876.000          CALL HLABEL(24,DNAMX)
1863 - 1877.000     50    TXT=STRTX-STIM(ISEGF)
1864 - 1878.000          XS1=XSTRT*((TXT/TOT(ISEGF))*XSIN)
1865 - 1879.000          XS2=XS1+.075
1866 - 1880.000          XS3=XS1-.075
1867 - 1881.000 C
1868 - 1882.000 C         PLOT TRIANGLE
1869 - 1883.000 C
1870 - 1884.000          CALL MOVEA(XS1,YS3)
```

```
1871 - 1885.000          CALL DRAWA(XS1,YS2)
1872 - 1886.000          CALL DRAWA(XS2,YS1)
1873 - 1887.000          CALL DRAWA(XS3,YS1)
1874 - 1888.000          CALL DRAWA(XS1,YS2)
1875 - 1889.000          RETURN
1876 - 1890.000          END
1877 - 1891.000           SUBROUTINE WAIT(I,ISIZE,ICPY)
1878 - 1892.000 C
1879 - 1893.000           CHARACTER*1 J
1880 - 1894.000 C
1881 - 1895.000           IF(ICPY.EQ.1) GO TO 150
1882 - 1896.000           READ(1,100)J
1883 - 1897.000    100    FORMAT(A1)
1884 - 1898.000           GO TO 200
1885 - 1899.000    150    CALL HDCOPY
1886 - 1900.000    200    IF(ISIZE .NE. 0) CALL CLEAR(ISIZE)
1887 - 1900.300           DO 300, IJ=1,10000
1888 - 1900.400    300    IJJ=1
1889 - 1901.000           I = 0
1890 - 1902.000 C
1891 - 1903.000 C         EXIT
1892 - 1904.000           RETURN
1893 - 1905.000 C
1894 - 1906.000           END
1895 - 1907.000           SUBROUTINE NTYKPS(IPASS)
1896 - 1908.000 C
1897 - 1909.000           INCLUDE MATRIX
1898 - 1910.000 C
1899 - 1911.000 C         RESET ARRAYS
1900 - 1912.000           DO 10 I = 1,NTASKS
1901 - 1913.000                ISCHED(I) = 0
1902 - 1914.000                MARK(I) = 0
1903 - 1915.000                WC(I) = 0
1904 - 1916.000                SC(I) = 0
1905 - 1917.000     10    CONTINUE
1906 - 1918.000           NMARKS = 0
1907 - 1919.000           WC(HEAD)= ISTRL(HEAD,WC(HEAD))
1908 - 1920.000           WC(HEAD) = ISTRR(HEAD,WC(HEAD))
1909 - 1921.000           SC(HEAD) = WC(HEAD)
1910 - 1922.000 C
1911 - 1923.000 C         IF PASS IS FORWARDS
1912 - 1924.000           IF(IPASS .NE. EARLY) GO TO 12
1913 - 1925.000 C
1914 - 1926.000 C              PUT  FIRST TASK WITH NO PREDECESSORS ON CHAIN     WC
1915 - 1927.000                DO 13 I = 1,NTASKS
1916 - 1928.000                     IF(NPRED(I) .EQ. 0) GO TO 14
1917 - 1929.000     13         CONTINUE
1918 - 1930.000     14         CONTINUE
1919 - 1931.000                CALL INSERT(I,HEAD,WC)
1920 - 1932.000                GO TO 17
1921 - 1933.000 C
1922 - 1934.000 C              PUT FIRST TASK WITH NO SUCESSORS ON CHAIN WC
```

```
1923 - 1935.000     12          DO 15 I = 1,NTASKS
1924 - 1936.000                     IF(NSUCR(I) .EQ. 0) GO TO 16
1925 - 1937.000     15          CONTINUE
1926 - 1938.000     16          CONTINUE
1927 - 1939.000                 CALL INSERT(I,HEAD,WC)
1928 - 1940.000 C
1929 - 1941.000 C
1930 - 1942.000 C         LOOP UNTIL ALL TASKS HAVE BEEN MARKED
1931 - 1943.000     17    IDIR = IPASS
1932 - 1944.000     20    CONTINUE
1933 - 1945.000 C
1934 - 1946.000 C             IF THIS IS A FORWARD PASS
1935 - 1947.000               IF(IDIR.NE.EARLY)GO TO 25
1936 - 1948.000 C
1937 - 1949.000 C                 PROCESS A FORWARD PASS THROUGH NETWORK
1938 - 1950.000                   CALL WORK(EARLY,NSUCR,SUCR,NPRED)
1939 - 1951.000                   IDIR=LATE
1940 - 1952.000                   GO TO 30
1941 - 1953.000 C
1942 - 1954.000 C                 PROCESS A BACKWARD PASS THROUGH NETWORK
1943 - 1955.000     25            CALL WORK (LATE,NPRED,PRED,NSUCR)
1944 - 1956.000                   IDIR=EARLY
1945 - 1957.000     30    CONTINUE
1946 - 1958.000 C
1947 - 1959.000 C
1948 - 1960.000 C         END LOOP-UNTIL
1949 - 1961.000           IF(NMARKS.LT.NTASKS) GO TO 20
1950 - 1962.000 C
1951 - 1963.000           RETURN
1952 - 1964.000 C
1953 - 1965.000           END
1954 - 1966.000       SUBROUTINE SCLY(INMEN)
1955 - 1967.000 C
1956 - 1968.000 C     THIS ROUTINE DETERMINES SCALE OF Y-AXIS ON RESOURCE GRAPHS
1957 - 1969.000 C
1958 - 1970.000    10 CONTINUE
1959 - 1971.000       IDIV=INMEN/5
1960 - 1972.000       FIDIV=FLOAT(INMEN)/5.
1961 - 1973.000       FDIV=FLOAT(INMEN)/5.
1962 - 1974.000       IF(FIDIV.EQ.FDIV)GO TO 20
1963 - 1975.000       INMEN=INMEN+1
1964 - 1976.000       GO TO 10
1965 - 1977.000    20 CONTINUE
1966 - 1978.000 C
1967 - 1979.000       RETURN
1968 - 1980.000       END
1969 - 1981.000       SUBROUTINE PRINET
1970 - 1982.000 C
1971 - 1983.000 C     THIS SUBROUTINE PRINTS THE NETWORK TASK DESCRIPTION IN WATERFALL ORDER
1972 - 1984.000 C     WITH EACH TASK FOLLOWED BY THE DESCRIPTIONS OF ITS PREDECESSORS AND
1973 - 1985.000 C     SUCCESSORS
1974 - 1986.000 C
```

```
      INCLUDE MATRIX
      INCLUDE COP
C
C     INITIALIZE
      IPAGE = 0
      ITSK = IHALFR(WC(HEAD))
      IF(ITSK .EQ. NULL)GO TO 999
C
C     LOOP UNTIL ALL TASKS HAVE BEEN LISTED
   10 CONTINUE
C
C          PRINT PAGE HEADER
           CALL CLEAR(4)
           IPAGE = IPAGE + 1
           LINES = 0
           IOUTS=0
           PRINT 100, (FTITLE(I),I=1,6),IPAGE,ITUNIT
  100      FORMAT(//T22,'NETWORK DESCRIPTION'//T22,6A6,T60,'PAGE',I4//
     &          ' TASK',T43,'DURATION',T56,'START',T68,'STOP'/
     &          ' LABEL    TASK DESCRIPTION',T43,'(',A6,')',T56,
     &          2('TIME',8X)//)
C
C          LOOP WHILE PAGE IS NOT FULL
   20      CONTINUE
           NLINES = LINES + NPRED(ITSK) + NSUCR(ITSK) + 6
           IF(NLINES .GT. 59) GO TO 30
C
C               PRINT TASK
                ETIME = SDUR(ITSK) + STIME(ITSK)
                IF(ISEG.EQ.1)CALL CRTSKN(TASK(ITSK),IQT,IQUTS,
     *          IORLPF,IORDRL)
                IF(IQT.EQ.1)GO TO 40
                PRINT 200, TASK(ITSK),(TSKLBL(ITSK,J),J=1,4),
     &               SDUR(ITSK),STIME(ITSK),ETIME
  200           FORMAT(1X,A4,4X,4A6,3(5X,F8.1)//)
C
C               IF TASK HAS NO PREDECESSORS
                IF(NPRED(ITSK) .EQ. 0)WRITE(6,213)
  213           FORMAT(T12,'NO PREDECESSORS'/)
                IF(NPRED(ITSK) .NE. 0) GO TO 215
                GO TO 214
C
C                    ELSE PRINT PREDECESSORS
  215               WRITE(6,216)
  216               FORMAT(T12,'PREDECESSORS:'/)
                   DO 21 I = 1, NPRED(ITSK)
                        ITSKK = PRED(ITSK,I)
                        WRITE(6,211)(TSKLBL(ITSKK,J),J=1,4)
  211                   FORMAT(T15,4A6)
   21              CONTINUE
                   WRITE(6,212)
  212              FORMAT(/)
```

```
2027 - 2039.000 C
2028 - 2040.000  214            CONTINUE
2029 - 2041.000 C
2030 - 2042.000 C               IF TASK HAS NO SUCCESSORS
2031 - 2043.000                 IF(NSUCR(ITSK) .EQ. 0)WRITE(6,221)
2032 - 2044.000  221            FORMAT(T12,'NO SUCCESSORS'/)
2033 - 2045.000                 IF(NSUCR(ITSK) .NE. 0) GO TO 223
2034 - 2046.000                 GO TO 222
2035 - 2047.000 C
2036 - 2048.000 C                    ELSE LIST SUCCESSORS
2037 - 2049.000  223                 WRITE(6,220)
2038 - 2050.000  220                 FORMAT(T12,'SUCCESSORS:'/)
2039 - 2051.000                      DO 22 I = 1,NSUCR(ITSK)
2040 - 2052.000                           ITSKK = SUCR(ITSK,I)
2041 - 2053.000                           WRITE(6,211)(TSKLBL(ITSKK,J),J=1,4)
2042 - 2054.000   22                   CONTINUE
2043 - 2055.000                      WRITE(6,212)
2044 - 2056.000 C
2045 - 2057.000  222            CONTINUE
2046 - 2058.000 C
2047 - 2059.000 C          END OF TASK PRINT LOOP
2048 - 2060.000            LINES = NLINES
2049 - 2061.000    40      ITSK = IHALFR(NC(ITSK))
2050 - 2062.000            IF(ITSK .NE. NULL) GO TO 20
2051 - 2063.000 C
2052 - 2064.000 C          FULL PAGE - WAIT
2053 - 2065.000   30       CONTINUE
2054 - 2066.000            CALL WAIT(1,4,ICPY)
2055 - 2067.000 C
2056 - 2068.000 C     END LOOP - CHECK IF ALL TASKS HAVE BEEN PRINTED
2057 - 2069.000       IF(ITSK .NE. NULL) GO TO 10
2058 - 2070.000 C
2059 - 2071.000  999  CONTINUE
2060 - 2072.000       RETURN
2061 - 2073.000 C
2062 - 2074.000       END
2063 - 2075.000       FUNCTION IHALFL(IARG)
2064 - 2076.000       LARG=IARG
2065 - 2077.000       IHALFL=ISL(LARG,-16)
2066 - 2078.000       RETURN
2067 - 2079.000       END
2068 - 2080.000       FUNCTION IHALFR(IARG)
2069 - 2081.000       LARG=IARG
2070 - 2082.000       IHALFR=ISL(LARG,16)
2071 - 2083.000       IHALFR=ISL(IHALFR,-16)
2072 - 2084.000       RETURN
2073 - 2085.000       END
2074 - 2086.000       FUNCTION ISTRL(IFM,ITO)
2075 - 2087.000       LIFM=IFM
2076 - 2088.000       LITO=ITO
2077 - 2089.000       LIFM=ISL(LIFM,16)
2078 - 2090.000       LITO=ISL(LITO,16)
```

```
2079 - 2091.000        LITO=ISL(LITO,-16)
2080 - 2092.000        ISTRL=LITO+LIFM
2081 - 2093.000        RETURN
2082 - 2094.000        END
2083 - 2095.000        FUNCTION ISTRR(IFM,ITO)
2084 - 2096.000        LIFM=IFM
2085 - 2097.000        LITO=ITO
2086 - 2098.000        LIFM=ISL(LIFM,16)
2087 - 2099.000        LIFM=ISL(LIFM,-16)
2088 - 2100.000        LITO=ISL(LITO,-16)
2089 - 2101.000        LITO=ISL(LITO,16)
2090 - 2102.000        ISTRR=LITO+LIFM
2091 - 2103.000        RETURN
2092 - 2104.000        END
2093 - 2105.000           SUBROUTINE WORK(IMARK,NLINK,LIST,MLINK)
2094 - 2106.000 C
2095 - 2107.000           INCLUDE MATRIX
2096 - 2108.000           DIMENSION NLINK(1),LIST(MAXTSK,MAXLNK),MLINK(1)
2097 - 2109.000 C
2098 - 2110.000 C         DELETE ALL TASKS EXCEPT THOSE WITH NO PRED/SUCR
2099 - 2111.000           WCPTR=IHALFR(WC(HEAD))
2100 - 2112.000 C
2101 - 2113.000 C         LOOP-WHILE TASKS ON LIST
2102 - 2114.000     10    CONTINUE
2103 - 2115.000           IF(WCPTR.EQ.NULL)GO TO 19
2104 - 2116.000 C
2105 - 2117.000 C               IF THE NUMBER OF PRED/SUCR'S IS NOT ZERO
2106 - 2118.000                 IF(NLINK(WCPTR).EQ.0) GO TO 15
2107 - 2119.000 C
2108 - 2120.000 C                     THEN DELETE THIS TASK
2109 - 2121.000                       LINK2 = WCPTR
2110 - 2122.000                       WCPTR = IHALFL(WC(WCPTR))
2111 - 2123.000                       CALL DEL(LINK2,WC)
2112 - 2124.000 C
2113 - 2125.000 C                 INCREMENT POINTER TO NEXT TASK
2114 - 2126.000     15            CONTINUE
2115 - 2127.000                   WCPTR=IHALFR(WC(WCPTR))
2116 - 2128.000 C
2117 - 2129.000 C         END LOOP-WHILE
2118 - 2130.000           GO TO 10
2119 - 2131.000 C
2120 - 2132.000 C         INITIALIZE POINTER
2121 - 2133.000     19    CONTINUE
2122 - 2134.000           WCPTR=IHALFR(WC(HEAD))
2123 - 2135.000 C
2124 - 2136.000 C         LOOP-WHILE MORE TASKS ARE ON CHAIN
2125 - 2137.000     20    CONTINUE
2126 - 2138.000           IF (WCPTR.EQ.NULL)GO TO 30
2127 - 2139.000 C
2128 - 2140.000 C               IF THIS TASK IS NOT MARKED 'IMARK'
2129 - 2141.000                 IF(MARK(WCPTR).EQ.IMARK) GO TO 23
2130 - 2142.000 C
```

```
2131 - 2143.000 C                    THEN LOOP UNTIL ALL PRED/SUCR'S ARE MOVED TO END OF C . MAIN
2132 - 2144.000                      NL=NLINK(WCPTR)
2133 - 2145.000                      IF(NL.EQ.0)GO TO 23
2134 - 2146.000                      DO 22 I=1,NL
2135 - 2147.000 C
2136 - 2148.000 C                         IF PRED/SUCR IS ALREADY ON WC
2137 - 2149.000                           ILINK=LIST(WCPTR,I)
2138 - 2150.000                           IF(IHALFR(WC(ILINK)).EQ.0)GO TO 21
2139 - 2151.000 C
2140 - 2152.000 C                              THEN REMOVE IT FROM IT'S CURRENT POSITION
2141 - 2153.000                                CALL DEL(ILINK,WC)
2142 - 2154.000 C
2143 - 2155.000 C                         PLACE TASK AT END OF WC
2144 - 2156.000    21                     CONTINUE
2145 - 2157.000                           CALL INSERT(ILINK,HEAD,WC)
2146 - 2158.000 C
2147 - 2159.000 C                      END LOOP-UNTIL
2148 - 2160.000    22                  CONTINUE
2149 - 2161.000 C
2150 - 2162.000 C               IF THIS TASK IS MARKED
2151 - 2163.000    23           CONTINUE
2152 - 2164.000                 IF(MARK(WCPTR).EQ.0)GO TO 29
2153 - 2165.000 C
2154 - 2166.000 C                    THEN DELETE IT FROM THE WC
2155 - 2167.000                      LINK2 = WCPTR
2156 - 2168.000                      WCPTR = IHALFL(WC(WCPTR))
2157 - 2169.000                      CALL DEL(LINK2,WC)
2158 - 2170.000 C
2159 - 2171.000 C               INCREMENT POINTER
2160 - 2172.000    29           CONTINUE
2161 - 2173.000                 WCPTR = IHALFR(WC(WCPTR))
2162 - 2174.000 C
2163 - 2175.000 C          END LOOP-WHILE
2164 - 2176.000            GO TO 20
2165 - 2177.000 C
2166 - 2178.000 C          LOOP UNTIL ALL TASKS ON WC ARE MARKED
2167 - 2179.000    30      CONTINUE
2168 - 2180.000            WCPTR=IHALFR(WC(HEAD))
2169 - 2181.000    31      CONTINUE
2170 - 2182.000 C
2171 - 2183.000 C               MARK THIS TASK AND INCREMENT POINTER
2172 - 2184.000                 MARK(WCPTR)=IMARK
2173 - 2185.000                 NMARKS=NMARKS+1
2174 - 2186.000                 WCPTR=IHALFR(WC(WCPTR))
2175 - 2187.000 C
2176 - 2188.000 C          END LOOP
2177 - 2189.000            IF(WCPTR.NE.NULL)GO TO 31
2178 - 2190.000 C
2179 - 2191.000 C          COPY WC TO SC
2180 - 2192.000            CALL COPY(WC,SC)
2181 - 2193.000 C
2182 - 2194.000 C          EXIT
```

```
2183 - 2195.000    999    CONTINUE
2184 - 2196.000           RETURN
2185 - 2197.000 C
2186 - 2198.000           END
2187 - 2199.000           SUBROUTINE SEGMNT
2188 - 2200.000           INCLUDE MATRIX
2189 - 2201.000 C
2190 - 2202.000           WORDRF=0.
2191 - 2203.000           WORDRL=0.
2192 - 2204.000 C
2193 - 2205.000           WRITE(1,5)SEGMT(ISEG)
2194 - 2206.000      5    FORMAT(' THE PLOT AND TABLE SEGMENTATION IS CURRENTLY ',A3)
2195 - 2207.000           WRITE(1,6)
2196 - 2208.000      6    FORMAT(' INPUT A 1 IF YOU WISH IT ON OR A 2 IF '
2197 - 2209.000        *  ' YOU WISH IT OFF')
2198 - 2210.000           READ(1,7)ISEG
2199 - 2211.000      7    FORMAT(I1)
2200 - 2212.000           IF(ISEG.EQ.2) GO TO 50
2201 - 2213.000 C
2202 - 2214.000 C          SET TASKS AND TIMES FOR SEGMENTATION
2203 - 2215.000           WRITE(1,10)
2204 - 2216.000     10    FORMAT(' FOR INPUT-ORDER REPORTS AND PLOTS,'/
2205 - 2217.000        *  ' INPUT THE FIRST TASK IN THE SEQUENCE TO BE DISPLAYED')
2206 - 2218.000           READ(1,11)IORDRF
2207 - 2219.000     11    FORMAT(A4)
2208 - 2220.000           WRITE(1,20)
2209 - 2221.000     20    FORMAT(' INPUT THE LAST TASK IN THE SEQUENCE TO BE DISPLAYED')
2210 - 2222.000           READ(1,11)IORDRL
2211 - 2223.000           WRITE(1,30)
2212 - 2224.000     30    FORMAT(' FOR WATERFALL ORDER REPORTS AND PLOTS,'/
2213 - 2225.000        *  ' INPUT THE EARLIEST START TIME OF INTEREST')
2214 - 2226.000           READ(1,31)WORDRF
2215 - 2227.000     31    FORMAT(F6.0)
2216 - 2228.000           WRITE(1,40)
2217 - 2229.000     40    FORMAT(' INPUT THE LATEST START TIME OF INTEREST')
2218 - 2230.000           READ(1,31)WORDRL
2219 - 2231.000 C
2220 - 2232.000     50    CONTINUE
2221 - 2233.000           RETURN
2222 - 2234.000           END
2223 - 2235.000           SUBROUTINE CKTSKN(TASK,IQT,IQUTS,IORDRF,IORDRL)
2224 - 2236.000           INTEGER TASK
2225 - 2237.000 C
2226 - 2238.000 C          THIS ROUTINE DETERMINES IF TASK IS WITHIN THE SEGMENT
2227 - 2239.000 C          BOUNDRIES FOR TIMELINE TYPE TABLES
2228 - 2240.000           IF(IQUTS.EQ.2)IQUTS=3
2229 - 2241.000           IF(IORDRF.EQ.TASK)IQUTS=1
2230 - 2242.000           IF(IORDRL.EQ.TASK)IQUTS=2
2231 - 2243.000           IF(IQUTS.EQ.0)IQT=1
2232 - 2244.000           IF(IQUTS.EQ.1)IQT=0
2233 - 2245.000           IF(IQUTS.EQ.2)IQT=0
2234 - 2246.000           IF(IQUTS.EQ.3)IQT=1
```

```
2235 - 2247.000 C
2236 - 2248.000          RETURN
2237 - 2249.000          END
2238 - 2250.000          SUBROUTINE CKSTTM(STIME,IQT,TSTOP,WORDRF,WORDRL)
2239 - 2251.000 C
2240 - 2252.000 C        THIS ROUTINE CHECKS THE TIME TO DETERMINE IF TASK IS
2241 - 2253.000 C        COMPLETELY OUT OF THE BOUNDRIES OF THE SEGMENT
2242 - 2254.000 C        FOR WATERFALL TYPE TABLES
2243 - 2255.000          IQT=0
2244 - 2256.000          IF(TSTOP.LE.WORDRF)IQT=1
2245 - 2257.000          IF(STIME.GE.WORDRL)IQT=1
2246 - 2258.000 C
2247 - 2259.000          RETURN
2248 - 2260.000          END
2249 - 2261.000          SUBROUTINE CKHTAB(HTABLE,IS,IE,WORDRF,WORDRL)
2250 - 2262.000 C
2251 - 2263.000 C        THIS ROUTINE CHECKS THE TIMES ON THE HISTOGRAMS TO
2252 - 2264.000 C        DETERMINE IF IT IS WITHIN THE SEGMENT BOUNDRIES FOR
2253 - 2265.000 C        THE HISTOGRAM TABLE
2254 - 2266.000 C
2255 - 2267.000          DIMENSION HTABLE(10000)
2256 - 2268.000 C
2257 - 2269.000     10   CONTINUE
2258 - 2270.000          IF(HTABLE(IS).LT.WORDRF.AND.HTABLE(IS+2).LE.WORDRF)
2259 - 2271.000        * IS=IS+2;GO TO 10
2260 - 2272.000     20   CONTINUE
2261 - 2273.000          IF(HTABLE(IE-1).GE.WORDRL)IE=IE-2;GO TO 20
2262 - 2274.000 C
2263 - 2275.000          RETURN
2264 - 2276.000          END
2265 - 2277.000          SUBROUTINE CKSTMP(WORDRL,WORDRF,ISKP)
2266 - 2278.000          INCLUDE TPLOTS
2267 - 2279.000 C
2268 - 2280.000 C        THIS ROUTINE SET START AND END TIMES FOR TASKS WITHIN
2269 - 2281.000 C        SEGMENT BOUNDRIES FOR WATER FALL PLOTS
2270 - 2282.000          ISKP=0
2271 - 2283.000          IF(IPASS.NE.1)GO TO 10
2272 - 2284.000 C
2273 - 2285.000 C        WHEN TASK ON CRITICAL PATH
2274 - 2286.000          ENDT=STRT(1)+DURR(1)
2275 - 2287.000          IF(STRT(1).GE.WORDRL.OR.ENDT.LE.WORDRF)ISKP=1
2276 - 2288.000          IF(ISKP.EQ.1)GO TO 999
2277 - 2289.000          IF(STRT(1).LT.WORDRF)STRT(1)=WORDRF
2278 - 2290.000          IF(ENDT.GT.WORDRL)DURR(1)=WORDRL-STRT(1)
2279 - 2291.000          IF(ENDT.LE.WORDRL.AND.DURR(1).NE.0)DURR(1)=ENDT-STRT(1)
2280 - 2292.000          GO TO 999
2281 - 2293.000 C
2282 - 2294.000 C        WHEN TASK NOT ON CRITICAL PATH
2283 - 2295.000     10   CONTINUE
2284 - 2296.000          IF(IFLG(1).EQ.1)GO TO 20
2285 - 2297.000 C
2286 - 2298.000 C        WHEN JOB SCHEDULED EARLY
```

```
2287 - 2299.000 C
2288 - 2300.000 C              FOR ACTUAL TIME AND DURATION
2289 - 2301.000                ENDTT=STRT(2)+DURR(2)
2290 - 2302.000                ENDT=STRT(1)+DURR(1)
2291 - 2303.000                IF(STRT(1).GE.WORDRL.OR.ENDTT.LE.WORDRF)ISKP=1
2292 - 2304.000                IF(ISKP.EQ.1)GO TO 999
2293 - 2305.000                IF(STRT(1).LT.WORDRF)STRT(1)=WORDRF
2294 - 2306.000                IF(ENDT.LT.WORDRF)DURR(1)=0
2295 - 2307.000                IF(ENDT.GT.WORDRL)DURR(1)=WORDRL-STRT(1)
2296 - 2308.000                IF(ENDT.LE.WORDRL.AND.DURR(1).NE.0)DURR(1)=ENDT-STRT(1)
2297 - 2309.000 C
2298 - 2310.000 C              FOR SLACK TIME
2299 - 2311.000                IF(STRT(2).LT.WORDRF)STRT(2)=WORDRF
2300 - 2312.000                IF(STRT(2).GT.WORDRL)STRT(2)=WORDRL
2301 - 2313.000                IF(ENDTT.GT.WORDRL)DURR(2)=WORDRL-STRT(2)
2302 - 2314.000                IF(ENDTT.LE.WORDRL.AND.DURR(2).NE.0)
2303 - 2315.000       *        DURR(2)=ENDTT-STRT(2)
2304 - 2316.000                GO TO 999
2305 - 2317.000 C
2306 - 2318.000 C              WHEN JOB SCHEDULED LATE
2307 - 2319.000     20         CONTINUE
2308 - 2320.000 C
2309 - 2321.000 C              FOR ACTUAL TIME AND DURATION
2310 - 2322.000                ENDT=STRT(2)+DURR(2)
2311 - 2323.000                IF(STRT(1).GE.WORDRL.OR.ENDT.LE.WORDRF)ISKP=1
2312 - 2324.000                IF(ISKP.EQ.1)GO TO 999
2313 - 2325.000                IF(STRT(2).LT.WORDRF)STRT(2)=WORDRF
2314 - 2326.000                IF(STRT(2).GT.WORDRL)STRT(2)=WORDRL
2315 - 2327.000                IF(ENDT.GT.WORDRL)DURR(2)=WORDRL-STRT(2)
2316 - 2328.000                IF(ENDT.LE.WORDRL.AND.DURR(2).NE.0)DURR(2)=ENDT-STRT(2)
2317 - 2329.000 C
2318 - 2330.000 C              FOR SLACK TIME
2319 - 2331.000                ENDT=STRT(1)+DURR(1)
2320 - 2332.000                IF(STRT(1).LT.WORDRF)STRT(1)=WORDRF
2321 - 2333.000                IF(ENDT.LT.WORDRF)DURR(1)=0.
2322 - 2334.000                IF(ENDT.GT.WORDRL)DURR(1)=WORDRL-STRT(1)
2323 - 2335.000                IF(ENDT.LE.WORDRL.AND.DURR(1).NE.0)DURR(1)=ENDT-STRT(1)
2324 - 2336.000 C
2325 - 2337.000    999         CONTINUE
2326 - 2338.000 C
2327 - 2339.000                RETURN
2328 - 2340.000                END
2329 - 2341.000                SUBROUTINE CKHIST(TIMES,TIMES2,ISET,NLVL,J)
2330 - 2342.000 C
2331 - 2343.000 C              THIS ROUTINE DETERMINES WHICH PARTS
2332 - 2344.000 C              OF THE HISTOGRAM PLOT BELONG IN THE SEGMENT
2333 - 2345.000                INCLUDE MATRIX
2334 - 2346.000 C
2335 - 2347.000                IF(TIMES.LT.WORDRF.AND.TIMES2.GT.WORDRF)
2336 - 2348.000       *        ISET=2;TIMES=WORDRF
2337 - 2349.000                IF(TIMES.LT.WORDRF.AND.TIMES2.LE.WORDRF)ISET=1
2338 - 2350.000                IF(TIMES.GT.WORDRL)ISET=2;TIMES=WORDRL
```

```
2338 - 2351.000           IF(TIMES.LT.WORDRF.AND.NLVL.EQ.J)TIMES=WORDRF;ISET=2
2340 - 2352.000 C
2341 - 2353.000           RETURN
2342 - 2354.000           END
2343 - 2355.000           SUBROUTINE TIMLIN(NN,IPCESS)
2344 - 2356.000 C
2345 - 2357.000           INCLUDE MATRIX
2346 - 2358.000           INCLUDE TPLOTS
2347 - 2359.000           INCLUDE CCP
2348 - 2360.000          COMMON/QSEG/ISEGF
2349 - 2361.000          COMMON/FLAG/IZB
2350 - 2362.000          DIMENSION ICODE(39)
2351 - 2363.000          DATA IDTES/'D'/
2352 - 2364.000          DATA IETES/'E'/
2353 - 2365.000 C
2354 - 2366.000 C        DETERMINE IF DATA HAS BEEN PROCESSED
2355 - 2367.000          IF(IPCESS.EQ.0)PRINT 2;CALL WAIT(I,0,0);GO TO 999
2356 - 2368.000        2 FORMAT(' DATA HAS NOT BEEN PROCESSED,HIT RETURN'/
2357 - 2369.000         * ' AND CHOOSE A ''3'' ON THE MENU')
2358 - 2370.000 C
2359 - 2371.000 C        DETERMINE IF THERE ARE ERRORS IN INPUT
2360 - 2372.000          IF(IFATAL.EQ.1)PRINT 3;CALL WAIT(I,0,0);GO TO 999
2361 - 2373.000        3 FORMAT(' THERE ARE ERRORS IN INPUT, HIT RETURN AND'/
2362 - 2374.000         * ' CHOOSE A ''6'' ON THE MENU')
2363 - 2375.000 C
2364 - 2376.000          ISEGF=1
2365 - 2377.000          IZB = 1
2366 - 2378.000 C
2367 - 2379.000 C           DETERMINE PLOT REQUIREMENTS
2368 - 2380.000        1    CONTINUE
2369 - 2381.000 C
2370 - 2382.000 C           SET TITLE AND TIME UNITS
2371 - 2383.000             DO 10 I = 1,6
2372 - 2384.000             ITITLE(I) = FTITLE(I)
2373 - 2385.000       10    CONTINUE
2374 - 2386.000             IUNITS = ITUNIT
2375 - 2387.000 C
2376 - 2388.000 C           DEFINE WINDOW IN INCHES
2377 - 2389.000             CALL VWINDC(0.,14.5,0.,10.9)
2378 - 2390.000             CALL CHRSIZ(3)
2379 - 2391.000             CALL NEWPAG
2380 - 2392.000 C
2381 - 2393.000 C           COMPUTE TIME SCALE FOR INPUT ORDER PLOTS OR ALL PLOTS WHEN
2382 - 2394.000 C           NOT SEGMENTED
2383 - 2395.000             I = IHALFR(LC(LCHEAD))
2384 - 2396.000             STIM(1) = CTIME(I)
2385 - 2397.000             I = IHALFL(LC(LCHEAD))
2386 - 2398.000             ENDTM(1) = CTIME(I)
2387 - 2399.000             TOT(1) = ENDTM(1) - STIM(1)
2388 - 2400.000 C
2389 - 2401.000 C           COMPUTE X-AXIS SCALE
2390 - 2402.000             XS = TOT(1)/XSIN
```

```
2391 - 2403.000           ISCALX = XS
2392 - 2404.000           IT = 0
2393 - 2405.000      20        CONTINUE
2394 - 2406.000                IF(ISCALX .LE. 10) GO TO 30
2395 - 2407.000                IT = IT + 1
2396 - 2408.000                ISCALX = ISCALX/10
2397 - 2409.000                GO TO 20
2398 - 2410.000      30   CONTINUE
2399 - 2411.000           XSCALE(1) = ISCALX*10**IT
2400 - 2412.000           IF(XSCALE(1).EQ.0)XSCALE(1)=1
2401 - 2413.000           XTICSP(1) = XSCALE(1)/XS
2402 - 2414.000           IF(ISEG.NE.1)GO TO 6
2403 - 2415.000 C
2404 - 2416.000 C         COMPUTE TIME SCALE FOR WATERFALL ORDER PLOTS WHEN SEGMENTED
2405 - 2417.000           STIM(2)=WORDRF
2406 - 2418.000           ENDTM(2)=WORDRL
2407 - 2419.000           TOT(2)=ENDTM(2)-STIM(2)
2408 - 2420.000 C
2409 - 2421.000 C         COMPUTE X-AXIS SCALE
2410 - 2422.000           XS=TOT(2)/XSIM
2411 - 2423.000           ISCALX=XS
2412 - 2424.000           IT=0
2413 - 2425.000      40        CONTINUE
2414 - 2426.000                IF(ISCALX.LE.10)GO TO 31
2415 - 2427.000                IT=IT+1
2416 - 2428.000                ISCALX=ISCALX/10
2417 - 2429.000                GO TO 40
2418 - 2430.000      31   CONTINUE
2419 - 2431.000           XSCALE(2)=ISCALX*10**IT
2420 - 2432.000           IF(XSCALE(2).EQ.9)XSCALE(2)=1
2421 - 2433.000           XTICSP(2)=XSCALE(2)/XS
2422 - 2434.000 C
2423 - 2435.000 C         DRAW AND LABEL FRAME
2424 - 2436.000       6   IF((MM.EQ.14.OR.MM.EQ.16.OR.MM.EQ.18).AND.ISEG.EQ.1)ISEGF=2
2425 - 2437.000           IF(MM.NE.13)CALL FRAME
2426 - 2438.000 C
2427 - 2439.000 C         SET UP AND PLOT TASK TIMELINE BARS
2428 - 2440.000           JOB = 1
2429 - 2441.000           NT = 1
2430 - 2442.000           WCPTR = IHALFR(WC(HEAD))
2431 - 2443.000           IF(WCPTR .EQ. NULL) GO TO 999
2432 - 2444.000 C
2433 - 2445.000 C         DETERMINE WHICH PLOT YOU WOULD LIKE
2434 - 2446.000           IF(MM.NE.14.AND.MM.NE.16.AND.MM.NE.18)GO TO 777
2435 - 2447.000 C
2436 - 2448.000 C         MAKE SURE IF SEGMENT FLAG IS ON INFORMATION IS GIVEN
2437 - 2449.000           IF(ISEG.EQ.1.AND.WORDRL.EQ.0.0)WRITE(1,4);
2438 - 2450.000       *   CALL WAIT(1,0,0);GO TO 777
2439 - 2451.000       4   FORMAT(' BEGINNING AND ENDING START TIMES WERE NOT GIVEN',
2440 - 2452.000       *     ' AND SEGMENT FLAG IS TURNED ON')
2441 - 2453.000 C
2442 - 2454.000 C         PLOT IN WATERFALL ORDER
```

```
2443 - 2455.000     50    CONTINUE
2444 - 2456.000           IF(IPFLG(WCPTR) .EQ. IDTES) GO TO 50
2445 - 2457.000           DO 51 J = 1,4
2446 - 2458.000           NAME(J) = TSKLBL(WCPTR,J)
2447 - 2459.000     51    CONTINUE
2448 - 2460.000           IF(SLACK(WCPTR).LT.-.0001.OR.SLACK(WCPTR).GT..0001)GOTO 55
2449 - 2461.000 C
2450 - 2462.000 C         CRITICAL PATH ACTIVITY
2451 - 2463.000           IPASS = 1
2452 - 2464.000           IFLG(1) = 3
2453 - 2465.000           IF(IPFLG(WCPTR) .EQ. IETES)IFLG(3) = 4
2454 - 2466.000           STRT(1) = STIME(WCPTR)
2455 - 2467.000           DURR(1) = SDUR(WCPTR)
2456 - 2468.000           GO TO 57
2457 - 2469.000 C
2458 - 2470.000 C         JOB IS NOT CRITICAL
2459 - 2471.000     55    CONTINUE
2460 - 2472.000           IF(MARK(WCPTR) .EQ. LATE) GO TO 56
2461 - 2473.000 C
2462 - 2474.000 C         EARLY SCHEDULED JOB
2463 - 2475.000           IPASS = 2
2464 - 2476.000           IFLG(1) = 2
2465 - 2477.000           STRT(1) = STIME(WCPTR)
2466 - 2478.000           DURR(1) = SDUR(WCPTR)
2467 - 2479.000           IFLG(2)=1
2468 - 2480.000           STRT(2) = STRT(1) + DURR(1)
2469 - 2481.000           DURR(2) = SLACK(WCPTR)
2470 - 2482.000           GO TO 57
2471 - 2483.000 C
2472 - 2484.000 C         JOB IS SCHEDULED LATE
2473 - 2485.000     56    CONTINUE
2474 - 2486.000           IPASS = 2
2475 - 2487.000           IFLG(2) = 2
2476 - 2488.000           STRT(2) = STIME(WCPTR)
2477 - 2489.000           DURR(2) = SDUR(WCPTR)
2478 - 2490.000           IFLG(1) = 1
2479 - 2491.000           DURR(1) = SLACK(WCPTR)
2480 - 2492.000           STRT(1) = STRT(2) - DURR(1)
2481 - 2493.000           GO TO 57
2482 - 2494.000 C
2483 - 2495.000 C         PLOT BARS
2484 - 2496.000     57    CONTINUE
2485 - 2497.000           ISKP=0
2486 - 2498.000           IF(ISEG.EQ.1)CALL CK3TMP(WORDRL,WOPDRF,ISKP)
2487 - 2499.000           IF(ISKP.EQ.1)GO TO 7
2488 - 2500.000                CALL BAR(MARK(WCPTR),EARLY,LATE)
2489 - 2501.000      7         WCPTR = IHALFR(WC(WCPTR))
2490 - 2502.000                IF(WCPTR .EQ. NULL) GO TO 889
2491 - 2503.000                GO TO 50
2492 - 2504.000 C
2493 - 2505.000 C     WATERFALL PLOTS COMPLETED
2494 - 2506.000    889 ILINE=41
```

```
2495 - 2507.000        CALL MOVEA(.2,.2)
2496 - 2508.000        CALL TSEND
2497 - 2509.000        CALL WAIT(I,3,ICPY)
2498 - 2510.000 C
2499 - 2511.000 C          PLOT IN TIMELINE ORDER
2500 - 2512.000    777     CONTINUE
2501 - 2513.000         IF(MM.NE.13.AND.MM.NE.16.AND.MM.NE.18)GO TO 999
2502 - 2514.000 C
2503 - 2515.000 C       MAKE SURE IF SEGMENT FLAG IS TURNED ON INFORMATION IS GIVEN
2504 - 2516.000         IF(ISEG.EQ.1.AND.IORDRF.EQ.BLANK)WRITE(1,5);
2505 - 2517.000        *CALL WAIT(I,0,0);GO TO 999
2506 - 2518.000       5 FORMAT(' BEGINNING AND ENDING TASK NUMBERS WERE NOT GIVEN',
2507 - 2519.000        * ' AND SEGMENT FLAG IS TURNED ON')
2508 - 2520.000 C
2509 - 2521.000 C       IF NOT GOING THROUGH TIMLIN, DEFINE WINDOW IN INCHES
2510 - 2522.000         IF(MM.NE.16.AND.MM.NE.18)GO TO 779
2511 - 2523.000         CALL VWINDO(0.,10.5,0.,10.9)
2512 - 2524.000         CALL CHRSIZ(3)
2513 - 2525.000         CALL NEWPAG
2514 - 2526.000         ISEGF=1
2515 - 2527.000         CALL FRAME
2516 - 2528.000 C
2517 - 2529.000    779     CONTINUE
2518 - 2530.000            IQT=0
2519 - 2531.000            IQUTS=0
2520 - 2532.000            DO 778 KK=1,NTASKS
2521 - 2533.000         IF(ISEG.EQ.1)CALL CKTSKN(TASK(KK),IQT,IQUTS,IORDRF,IORDRL)
2522 - 2534.000         IF(IQT.EQ.1)GO TO 778
2523 - 2535.000         CALL SETUP(KK)
2524 - 2536.000         CALL BAR(MARK(WCPTR),EARLY,LATE)
2525 - 2537.000     778 CONTINUE
2526 - 2538.000 C
2527 - 2539.000 C          TIMELINE PLOTS COMPLETED
2528 - 2540.000    888     CONTINUE
2529 - 2541.000            ILINE = 41
2530 - 2542.000            CALL MOVEA(.2,.2)
2531 - 2543.000            CALL TSEND
2532 - 2544.000            CALL WAIT(I,3,ICPY)
2533 - 2545.000 C
2534 - 2546.000 C          EXIT
2535 - 2547.000    999     CONTINUE
2536 - 2548.000            RETURN
2537 - 2549.000 C
2538 - 2550.000            END
2539 - 2551.000            SUBROUTINE FRAME
2540 - 2552.000 C
2541 - 2553.000            INCLUDE TPLOTS
2542 - 2554.000            INCLUDE MATRIX
2543 - 2555.000            COMMON/QSEG/ISEGF
2544 - 2556.000            CHARACTER*1 IBLNK
2545 - 2557.000            CHARACTER*36 DTITLE
2546 - 2558.000            INTEGER ITEXT(2),JTEXT(8),IUNITX(8),IUNIS(2)
```

```
2547 - 2559.000           INTEGER DTITL(9),DTITLX(36),IEN(5),IENX(20)
2548 - 2560.000           DIMENSION Y(3),XSC(8),YSC(8),IEND(1),YY(2)
2549 - 2561.000 C
2550 - 2562.000 C
2551 - 2563.000          DATA YY/10.5,10.81/
2552 - 2564.000          DATA Y/.9,.81,.65/
2553 - 2565.000          DATA XSC/0.,0.,14.5,14.5,0.,2.9275,2.9275,14.5/
2554 - 2566.000          DATA YSC/0.,10.9,10.9,0.,0.,10.9,.95,.95/
2555 - 2567.000 C
2556 - 2568.000 C        DEFINE WINDOW IN INCHES
2557 - 2569.000          CALL VWINDO(0.,14.5,0.,10.9)
2558 - 2570.000          CALL CHRSIZ(3)
2559 - 2571.000          CALL NEWPAG
2560 - 2573.000 C
2561 - 2574.000             DTITLE(:6)=ITITLE(1)
2562 - 2575.000             DTITLE(7:12)=ITITLE(2)
2563 - 2576.000             DTITLE(13:18)=ITITLE(3)
2564 - 2577.000             DTITLE(19:24)=ITITLE(4)
2565 - 2578.000             DTITLE(25:30)=ITITLE(5)
2566 - 2579.000             DTITLE(31:36)=ITITLE(6)
2567 - 2580.000             IBLNK=' '
2568 - 2581.000          X=2.9275
2569 - 2582.000          FNUM=STIN(ISEGF)+TES2
2570 - 2583.000          NTIC=(XSIN/XTICSP(ISEGF))+3.
2571 - 2584.000          XTIC1=XTICSP(ISEGF)
2572 - 2585.000 C
2573 - 2586.000 C          DRAW FRAME
2574 - 2587.000 C
2575 - 2588.000             CALL MOVEA(XSC(1),YSC(1))
2576 - 2589.000             DO 20 I=2,5
2577 - 2590.000       20     CALL DRAWA(XSC(I),YSC(I))
2578 - 2591.000             CALL MOVEA(XSC(6),YSC(6))
2579 - 2592.000             DO 25 I=7,8
2580 - 2593.000       25    CALL DRAWA(XSC(I),YSC(I))
2581 - 2594.000          DO 50 I=1,NTIC
2582 - 2595.000          IF(X.GT.14.06)GO TO 50
2583 - 2596.000 C
2584 - 2597.000 C        DRAW TIC MARK
2585 - 2598.000 C
2586 - 2599.000          CALL MOVEA(X,Y(1))
2587 - 2600.000          CALL DRAWA(X,Y(2))
2588 - 2601.000          CALL MOVEA(X,YY(1))
2589 - 2602.000          CALL DRAWA(X,YY(2))
2590 - 2603.000 C
2591 - 2604.000 C        PRINT CORRESPONDING VALUE
2592 - 2605.000 C
2593 - 2606.000          ENCODE(6,30,ITEXT)FNUM
2594 - 2607.000       30 FORMAT(F6.3)
2595 - 2608.000       40 X11=X-.33875
2596 - 2609.000          CALL KAN2AS(6,ITEXT,JTEXT)
2597 - 2610.000          CALL MOVEA(X11,Y(3))
2598 - 2611.000          CALL HLABEL(6,JTEXT)
```

```
2599 - 2612.000          FNUM=FNUM+XSCALE(ISEGF)
2600 - 2613.000          X=X+XTIC1
2601 - 2614.000      50    CONTINUE
2602 - 2615.000 C
2603 - 2616.000 C           PRINT UNITS LABEL
2604 - 2617.000 C
2605 - 2618.000             ENCODE(6,55,IUNIS)IUNITS
2606 - 2619.000      55     FORMAT(A6)
2607 - 2620.000             CALL KAN2AS(6,IUNIS,IUNITX)
2608 - 2621.000             CALL MOVEA(7.88984375,.45)
2609 - 2622.000             CALL HLABEL(6,IUNITX)
2610 - 2623.000 C
2611 - 2624.000 C           PRINT TITLE
2612 - 2625.000 C
2613 - 2626.000    200      CONTINUE
2614 - 2627.000             ENCODE(36,5,DTITL)DTITLE
2615 - 2628.000       5     FORMAT(A36)
2616 - 2629.000             CALL KAN2AS(36,DTITL,DTITLX)
2617 - 2630.000             CALL MOVEA(7.,.2)
2618 - 2631.000             CALL HLABEL(36,DTITLX)
2619 - 2632.000 C
2620 - 2633.000 C           PRINT DATE
2621 - 2634.000 C
2622 - 2635.000 C23         CALL SCLOCK(DATE,TIME,ESEC,F60SEC)
2623 - 2636.000 C           PRTDAT(1:2)=DATE(1:2)
2624 - 2637.000 C           PRTDAT(4:5)=DATE(3:4)
2625 - 2638.000 C           PRTDAT(9:10)=DATE(5:6)
2626 - 2639.000 C           PRTDAT(3:3)="/"
2627 - 2640.000 C           PRTDAT(6:6)="/"
2628 - 2641.000 C           CALL MOVEA(.2,.2)
2629 - 2642.000 C           CALL HLABEL(10,PRTDAT)
2630 - 2643.000 C
2631 - 2644.000 C           PRINT TOTAL TIME
2632 - 2645.000             ENCODE(18,210,IEN)ENDTM(ISEGF)
2633 - 2646.000    210      FORMAT("END TIME",F8.0)
2634 - 2647.000    211      FORMAT(A12)
2635 - 2648.000             CALL KAN2AS(18,IEN,IENX)
2636 - 2649.000             CALL MOVEA(12.,.44)
2637 - 2650.000             CALL HLABEL(18,IENX)
2638 - 2651.000 C
2639 - 2652.000 C           EXIT
2640 - 2653.000    999      CONTINUE
2641 - 2654.000             CALL TSEND
2642 - 2655.000           RETURN
2643 - 2656.000 C
2644 - 2657.000           END
2645 - 2658.000           FUNCTION RSCFCN(ITSK,IFCN,IRSC,TCOR)
2646 - 2659.000           RSCFCN=1.
2647 - 2660.000           RETURN
2648 - 2661.000           END
```

APPENDIX B

DOCUMENTATION CHECKLIST

DOCUMENTATION CHECKLIST

	Required	Completed	If not contained herein can be found in:
1. Title Page	✓	✓	
2. Table of Contents		✓	
3. Abstract	✓	✓	
4. Introduction			
a. Objective or Purpose	✓	✓	
b. MSFC Form 3559		✓	
c. Background		✓	
d. Related			
5. Problem Task Description		✓	
6. Method of Solution		✓	
7. Program Description			
a. Operating Environment			
(1) Hardware			
(a) Computer	✓	✓	
(b) Core Requirement	✓	✓	
(c) Magnetic Tapes	✓	✓	
(d) Card Punch	✓	✓	
(e) Plotter	✓	✓	
(f) Drum/Disc	✓	✓	
(g) Other	✓	✓	
(2) Software			
(a) Operating System	✓	✓	
(b) Programming Language(s)	✓	✓	
(c) Type of Run	✓	✓	
(d) Library Subroutines ..	✓	✓	
b. Program Specifications		✓	
c. Subroutines or Subprograms (other than library)		✓	
d. Source Code Listing	✓	✓	
e. Detailed Flow Charts			
8. Operating Instructions			
a. Deck Setup			
(1) Deck Sequence	✓	✓	
(2) Restart Sequence			
(3) Operator Instruction Card .			

(4) Magnetic Tape Save Labels .			
(5) Computer Time Requirements			
b. Input			
(1) Cards	✓	✓	
(2) Magnetic Tapes			
(3) Drum/Disc	✓	✓	
(4) Other	✓	✓	
c. Output			
(1) Cards			
(2) Magnetic Tapes			
(3) Drum/Disc			
(4) Printout		✓	
(5) Plots		✓	
(6) Other			
d. Restrictions and/or Limitations		✓	
e. Diagnostics		✓	
f. Test Case	✓	✓	
9. Symbols			
10. References			
11. Appendices			
a. Documentation Checklist	✓	✓	
b. Documentation Approval		✓	
c. Other			

APPENDIX C

DOCUMENTATION APPROVAL

DOCUMENTATION APPROVAL

Documentation Prepared By:

Jacqueline D Sessler
Signature

Jacque Sessler
Name (Typed)

Member of Technical Staff B
Title

6510
Organization

May 28, 1981/453-0918
Date and Telephone No.

Documentation Approved By:
(Supervisor of Person Preparing Documentation)

David R. Johnson
Signature

Dave Johnson
Name (Typed)

Manager
Title

6510
Organization

May 28, 1981/453-0620
Date and Telephone No.

Documentation Approved By:
Huntsville Computer Complex
(Project Officer or Monitor)

Van A. McAuley
Signature

Van McAuley
Name (Typed)

Mathematician
NASA Monitor
Title

AH33
Organization

6-3-81
May 28, 1981/453-2294
Date and Telephone No.

NASA—MSFC